
Spike Timing Dependent Adaptation: Minimising the Effect of Transistor Mismatch in an Analogue VLSI Neuromorphic System

Katherine Cameron



A thesis submitted for the degree of Doctor of Philosophy.
The University of Edinburgh.
September 2006



Abstract

Neuromorphic systems often call for subthreshold operation where transistor mismatch is a particular problem and this mismatch can affect the time constants of the design. This work is an investigation into whether Spike Timing Dependent Plasticity (STDP), a neural algorithm capable of adapting time delays within neural systems, can provide a method to minimise the effect of transistor mismatch.

This work is set within the context of a depth-from-motion algorithm, the performance of which will be degraded by mismatch when implemented in analogue VLSI. A circuit is designed which predicts the arrival of a spike from the timing of two earlier spikes. The error between the actual spike arrival time and the prediction is used to improve future predictions. Two spike timing dependent adaptation methods are described. These were fabricated using an Austria Microsystems (AMS) $0.35\mu\text{m}$ process and the results are reported. The key measure is the prediction error: before adaptation the error reflects the amount of mismatch within the prediction circuitry; after adaptation the error indicates to what extent the adaptive circuitry can minimise the effect of transistor mismatch. For both designs it is shown that the effect of transistor mismatch can be greatly reduced through spike timing dependent adaptation.

Declaration of originality

I hereby declare that:

- (a) the thesis has been composed entirely by myself;
- (b) the work is my own, except where clearly indicated, and originated in the School of Engineering and Electronics at The University of Edinburgh;
- (c) the work has not been submitted for any other degree or professional qualification.

Katherine Cameron

Acknowledgements

Many people have made the last four years great fun. These are just some of them.

Alan Murray as my supervisor provided me with advice and support.

Adria Bofill-i-Petit and Sebastian Loeda introduced me to the world of spiking neural networks and STDP.

Everyone involved in the EPSRC funded project *Silicon Spiking Systems: Collective Parallel Computation & Adaptive Sensory Systems*. The quarterly meetings always resulted in useful feedback about my work.

Alasdair Sutherland showed me how to use Cadence and David Stewart looked after its installation.

Vasin Boonsobhak worked on my first chip with me.

Patrice Fleury helped out with the layout of the pad ring for both my chips. Steve Collins and Robert Henderson both provided useful information when I didn't understand how it worked.

Bruce Rae, Will Holland, Amy Hartwell and Sarah Williams have all helped at some point along the line.

Elizabeth Rimmer provided some much needed proof reading. Paul Rimmer also, on occasion, read my work and provided constructive criticism on poster design and presentations.

A final thank you to George Cameron who said nice supportive things when required.

Contents

Declaration of originality	iii
Acknowledgements	iv
Contents	v
List of figures	vii
List of tables	ix
Acronyms and abbreviations	x
1 Introduction	1
1.1 Thesis Statement	2
1.2 Thesis Overview	3
2 Literature Review and Background	4
2.1 Introduction	4
2.2 Spike Timing and Temporal Codes	4
2.3 Hebbian Learning and Spike Timing Dependent Plasticity	7
2.4 Neuromorphic Circuitry	10
2.4.1 Silicon Neurons	10
2.4.2 Synapses	11
2.4.3 Connectivity	13
2.4.4 Applications and Learning Circuitry	13
2.5 Transistor Mismatch	16
2.6 Depth-from-Motion Algorithm	23
2.7 Summary and Conclusions	25
3 System Overview	27
3.1 Introduction	27
3.2 Integrate-and-Fire Neuron	27
3.3 Confirmation Block	28
3.4 Prediction Network	28
3.5 Adaptive Network	30
3.6 Summary	30
4 Prediction Circuit Architectures	31
4.1 Introduction	31
4.2 Two Capacitor Circuit	31
4.3 Dynamic Current Mirror	32
4.4 Circuitry suitable for spike timing dependent adaptation	35
4.5 Conclusions	37
5 Spike Timing Dependent Adaptation	39
5.1 Introduction	39
5.2 Adaptive Circuitry	39
5.3 Simulation Results	43

5.4	Chip Architecture	44
5.5	Chip Results	45
5.6	A Small Adaptive Pixel Array	46
5.7	Summary and Conclusions	49
6	An Improved Adaptive Technique	51
6.1	Introduction	51
6.2	The Redesigned Prediction Circuitry	51
6.3	Adaptive Circuitry	54
6.4	Chip Architecture	56
6.5	Results	56
6.5.1	Static Time Delays	57
6.5.2	Varying Inter-Spike Intervals	61
6.5.3	The Effect of Noisy Spikes on Adaptation	63
6.6	Summary and Conclusions	68
7	Summary and Conclusions	71
7.1	Work Undertaken	71
7.2	Future Work	72
7.3	Conclusions	73
A	Publications	74
B	Component Sizes and Bias Voltages	127
C	Control Circuits	136
	References	138

List of figures

2.1	Temporally Asymmetric Hebbian Learning algorithm	8
2.2	Current mirror integrator synapse	11
2.3	Common-Centroid Layouts	20
2.4	Dynamic Current Mirrors	21
2.5	Layout of the artificial retina	24
3.1	System Block Diagram	28
3.2	Leaky Integrate-and-Fire Neuron	29
3.3	Confirmation circuitry	29
3.4	Prediction network	30
4.1	Prediction circuit using two capacitors	32
4.2	Distribution of prediction errors for the two capacitor prediction circuitry	33
4.3	Dynamic current mirror circuit	34
4.4	Distribution of prediction errors for the dynamic current mirror prediction circuitry	35
4.5	Spike prediction circuit	36
4.6	Pre-adaptation waveforms	37
5.1	Pre- and post-adaptation results	40
5.2	Excitatory/inhibitory synapse circuit and I/V graph	41
5.3	Weight adaptation circuit	42
5.4	Weight adaptation algorithm	43
5.5	Distribution of prediction errors	44
5.6	Architecture of the first chip fabricated	45
5.7	Pre- and post-adaptation results	46
5.8	Oscilloscope trace showing weight change	47
5.9	Distribution of prediction errors	47
5.10	Distribution of prediction errors from pixel activated circuit	48
5.11	Distribution of prediction errors from the voltage controlled circuit	49
6.1	Spike prediction circuit	52
6.2	V_{slope} update circuitry	53
6.3	HiLo circuitry	54
6.4	The change in V_{slope} with prediction error	55
6.5	Architecture of the second chip fabricated	57
6.6	Prediction error during adaptation	58
6.7	Distribution of prediction errors - with adaptation	59
6.8	Distribution of prediction errors - dynamic current mirror	60
6.9	Distribution of prediction errors with varying V_{dd_mirr}	61
6.10	Prediction error during adaptation with varying inter-spike interval	64

6.11	Prediction error during adaptation with varying inter-spike interval, a comparison between different prediction methods	65
6.12	Prediction error during adaptation and the effect of noisy spikes	66
6.13	Prediction error and the effect of noisy spikes to the dynamic current mirror circuitry	67
6.14	Prediction error during adaptation while noisy spikes are applied	68
B.1	Standard Circuits	127
C.1	Prediction Circuit Control Circuitry	136
C.2	Weight Change Control Circuitry	137

List of tables

2.1	Rules for optimum matching	19
5.1	Mean and standard deviation of prediction error for design one	50
6.1	Mean and standard deviation of prediction error for design two	62
6.2	Delay times applied for variable delay tests	63

Acronyms and abbreviations

AER	Address Event Representation
AMS	Austria Microsystems
BiCMOS	A technology that includes both bipolar and MOS transistors
aVLSI	analogue Very Large Scale Integration
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
NMOS	N-type MOSFET
PMOS	P-type MOSFET
SPICE	Simulation Program with Integrated Circuits Emphasis
STDP	Spike Timing Dependent Plasticity
TAH	Temporally Asymmetric Hebbian
VLSI	Very Large Scale Integration

Chapter 1

Introduction

Traditionally it was thought that the important information carried by neural spikes was contained in the average firing rate of the neurons. This led to neural algorithms being mainly implemented as rate codes. As the importance of the timing of individual spikes became apparent [1], evidence from biological studies showed that the time interval between pre- and post-synaptic spikes could directly affect synaptic weight change. Markram et al. described in [2] that for tufted pyramidal cells in the neocortex, the synaptic weight could be increased (potentiated) if the pre-synaptic spike preceded the post-synaptic spike but was decreased (depressed) if the post-synaptic spike arrived first. Bi and Poo [3] and Zhang et al. [4] performed more detailed biological experiments which showed that the weight change did not depend just on the order of the pre- and post-synaptic spikes, but that the time interval between them affected the amount of weight change. Spike Timing Dependent Plasticity (STDP), as it came to be known, has been shown through simulation to be capable of improving synchronisation [5] and adapting synaptic delays [6–8] and, due to its relative simplicity, has been successfully implemented in analogue VLSI (aVLSI) [9, 10].

Pelgrom's law states that transistor mismatch is proportional to $1/WL$ [11]. Therefore, as transistor sizes decrease, mismatch will become an ever increasing problem, reducing circuit performance and decreasing fabrication yield. The design of neuromorphic circuits is inspired by the structure and principles of the biological nervous system, a system which is very low power and operates on long time constants. If analogue VLSI circuits are operated subthreshold in order to achieve one or both of these aims, transistor mismatch has a greater effect [12]. Mismatch can be reduced by increasing transistor area or by increasing gate-voltage overdrive, but other circuit constraints may restrict this [13]. Some compensation mechanisms have been proposed which have improved matching characteristics without having to increase individual transistor sizes. Dynamic techniques store bias voltages on capacitors connected to the gates of MOSFETs. These biases are usually set by an input current and then assumed constant between refreshes. They have allowed the fabrication of current mirrors, dividers and comparators [14–16] with improved accuracy. Dynamic storage requires periodic refreshes to replace charge that

has leaked away from the capacitor. Accurate current matching, without the need for refreshes, can be achieved using floating gate devices [17, 18]. Floating gate circuits¹ can be programmed with the bias voltage required to achieve a desired current. These devices are however far more sensitive to electrostatic discharge during handling [20] than ordinary circuits which can make them difficult to use. This work investigates an alternative approach. As transistor mismatch will affect the time constants in a design, and STDP has been shown to be capable of adapting delays, this work explores the possibility that spike timing dependent adaptation could reduce the effect of transistor mismatch within a neuromorphic system.

The neuromorphic system chosen as the context for this investigation was a depth-from-motion algorithm developed by Wörgötter et al. in 1999 [21]. It uses the relative timings of neural spikes to recover depth information from radial flow-fields. The algorithm was found to be susceptible to noise, which degraded the depth calculation. To improve noise rejection a prediction was added. This prediction uses the timing of two previous spikes to predict a third. If the third spike arrives within a time window around the prediction, it is accepted as genuine and the depth is calculated. If not, it is rejected as noise. If this prediction mechanism was implemented in aVLSI it would suffer from transistor mismatch. The result of this would be inaccurate predictions. The time difference between the actual and predicted spike could, if enough genuine predictions are generated, be used within a spike timing dependent algorithm to improve the prediction, and therefore minimise the effect of transistor mismatch.

1.1 Thesis Statement

This thesis presents work undertaken to examine whether spike timing dependent adaptation can be used to minimise the effect of mismatch within a neuromorphic system. This represents an attempt to apply a biological algorithm, that has proved powerful during simulation of neural systems, to an engineering problem.

¹A floating gate MOSFET is one in which the gate is surrounded on all sides by an insulator (typically Silicon dioxide)[19].

1.2 Thesis Overview

Chapter 2 is the literature review and provides background on spiking neurons, possible coding methods that use spike timing and Spike Timing Dependent Plasticity. Neuromorphic circuitry is then discussed as is transistor mismatch and the depth-from-motion algorithm.

Chapter 3 introduces the architecture of the system designed to implement the prediction element of the depth-from-motion algorithm.

Chapter 4 describes possible methods of making the prediction of spike 3. Three methods are described, and the levels of mismatch evaluated using Monte Carlo simulations. The first uses capacitor matching rather than current matching. The second uses dynamic biasing techniques to achieve matched currents. The third has a similar structure to the second but does not use dynamic biasing. This is the method to which spike timing dependent adaptation is applied. The two methods that do not use spike timing dependent adaptation are used as baselines which the adaptive circuitry must outperform.

Chapter 5 describes the first adaptive circuitry that was designed. Simulation and chip results are presented which show it to be capable of improving the accuracy of the prediction. Flaws with the implementation are highlighted, and chapter 6 describes improved circuitry designed to address them. Comparisons with a fabricated dynamic current mirror circuit are also presented.

Chapter 7 summarises the results, presents conclusions and discusses possible future work.

Chapter 2

Literature Review and Background

2.1 Introduction

As described in the introduction, this work investigates the use of spike timing dependent adaptation to minimise the effect of transistor mismatch within a neuromorphic VLSI system. This chapter provides a greater amount of information about spike timing, temporal codes and the spike timing dependent algorithms found in biology, together with accounts of computational simulations that show their power. Neuromorphic circuits are introduced, covering circuits from a silicon neuron up to learning circuits. There is a section on transistor mismatch which describes why it will become an ever greater problem as transistor sizes decrease, and provides examples of some methods used to minimise the effects. Finally the depth-from-motion algorithm is introduced to set the work in context.

2.2 Spike Timing and Temporal Codes

Much of the thought on neural coding was inspired by the work of E. G. Adrian [22–24] in which he observed that the rate of spiking increases as the input stimulus does. Rate coding as a measure of neural activity can take three forms, all of which have drawbacks. The first is a simple average of the number of spike firings during a time interval. To perform any sensible averaging, a number of spikes must have fired within the time interval. There are a number of biological studies which have shown that decisions are made with very few spikes, removing the ability to determine a firing rate. For example, flies can make course corrections in time scales during which only a few spikes have occurred [25].

The averaging could instead be done over a shorter period of time, but across a neural population. This is only appropriate if the neurons all perform a similar function. Alternatively the stimulus can be presented a number of times and the average result computed. This approach is unsuitable as the result is only correct if the internal state of the network has not been changed

by the previous presentations. More detail on these three methods can be found in [26] while rate codes are discussed in more detail in [1].

As rate codes can not explain all observed neural activity, the idea that the timing of individual spikes could be important needed to be investigated. In order to run simulations capable of including spike timing, spiking neuron models had to be developed [27]. These spiking neuron models are more biologically plausible than the McCulloch Pitts model or those with a sigmoid activation function. In addition Maass [28] showed that similar sized networks of spiking neurons could perform at least as well as networks of these other neurons; and in some cases outperform them. Spiking neurons have also been used to construct Hopfield networks [29], self-organising maps [30] and radial basis functions [31] or their temporal equivalents. Recurrent networks implementing winner-take-all functions have also been converted to use spiking neurons [32].

Beyond proving that spiking neural networks could perform traditional neural network tasks, thought was given to coding methods that could use spike timing. This led to simulations being run which showed that time dependent signals could be recovered from the spike times. For example, Bialek et al. [25] applied a time dependent signal to a wide-field, movement-sensitive neuron in the visual system of the blowfly. By recording the output spikes from the neuron they were able to recreate the input stimulus. Importantly, a very simple algorithm could be used to recover much of the information. In 1993 Gerstner et al. [33] showed that spike timing could hold information, in this case firing patterns, which was lost through rate coding methods, and that it could be retrieved after a learning process.

Although the temporal code could be recovered in [25] no attempt was made to suggest a coding method. Some of the possible contenders for the temporal code are time-to-first spike, phase and correlation coding.

The premise behind time-to-first spike coding is that as synaptic input is integrated on the neuron's membrane until the firing threshold is reached, the time between the stimulus onset and neural firing is dependent on stimulus strength. The greater the stimulus the faster the neuron will fire. While the actual firing time may hold important information it may be sufficient to know whether a neuron fired before or after another. If the firing times of neurons across a population are compared, with the neuron firing first ranked first etc., the resulting code is called rank order coding [34]. While this method does not preserve the actual timing of spikes,

it does encode their relationships. In [35] Van Rullen et al. employed a rank code on the output of a model of retinal ganglion cells. In this case the speed of firing was related to the contrast of the visual stimulus for that cell. Therefore the rank order code encodes the relative contrast but discards the absolute contrast. This is assumed to be acceptable as the visual system can function with varying background light intensities.

In order to perform the rank order decoding each rank had to be assigned a contrast value. To do this 3000 images were flashed before the model ganglion cells and the rank order code was noted. The average contrast normalised over the 3000 images was then calculated for each rank. From this attempts were made to perform image reconstruction. The results are impressive as the images were quite clear after only one percent of the neurons had fired.

One of the problems with this method is where to measure time-to-first spike from. In the work described in [35] the visual stimulus was flashed which provided an obvious initial time. Alternatively the time can be measured relative to a background oscillation, (phase coding), or from another spike, (synchrony or correlation coding).

Examples of phase coding have been observed in rat hippocampal cells [36] and in the cochlear nucleus of the barn owl [37]. O'Keefe and Recce [36] monitored the firing time of hippocampal place cells as rats ran along a track. Location affects the firing rate of the spikes, but in addition the phase relationship of the spikes to the hippocampal theta rhythm, a 7-12Hz sinusoidal oscillation visible in an electroencephalograph (EEG), changed. They showed that there was a better correlation between the rat's location and the phase relationship than between firing rate and position.

In Sullivan and Konishi [37] barn owls were exposed to sinusoidal tones and the firing of magnocellular neurons was shown to exhibit phase locking up to frequencies of 9kHz. If the frequency increased the phase delay also increased. The phase delay was relatively insensitive to sound intensity. Barn owls use interaural time differences to localise sounds in azimuth. In [38] a circuit for detecting interaural time differences was described. The first component was a phase locking circuit which preserved the timing information.

Evidence for synchronised or correlated spiking has been found. In [39] neurons from the frontal cortex of a rhesus monkey were recorded while the monkey performed a GO/NO-GO task. The temporal correlations between the neurons were different for the Go task from the NO-GO task. The average firing rate remained constant. Additionally in [40] the information

rate of a spike train was shown to increase when natural signals containing temporal correlations were encoded into spikes, therefore improving the coding efficiency of the spike train. Spike Timing Dependent Plasticity is a biological learning rule capable of detecting correlations and improving synchrony and is described in section 2.3.

2.3 Hebbian Learning and Spike Timing Dependent Plasticity

In the 1949 book ‘The Organization of Behaviour’ [41] Donald Hebb postulated that:

When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased.

From this postulate, Hebbian type learning came to mean that learning had to be local at the synapse, both pre- and post-synaptic activity had to be occurring, and some time dependent mechanism affected learning [42]. Hebbian learning has traditionally been implemented in response to the mean pre- and post-synaptic firing rate [43]. While in many cases this may be sufficient, it disregards the information contained within the actual spike-timing. In addition to showing that spike timing could hold information which was lost through rate coding methods, Gerstner et al. [33] showed that it was possible to use a Hebbian type learning algorithm to learn these patterns.

Hebb’s postulate implies some timing dependence between pre- and post-synaptic spikes. For cell A to take part in firing cell B the pre-synaptic spike must occur before the post-synaptic. In 1997 Markram et al. [2] showed that when the pre-synaptic spike occurred 10ms before the post-synaptic in pyramidal neurons, the synapse was potentiated. When the order was reversed, depression was induced. If the time difference was increased to 100ms, no change was observed.

In 1998 Bi and Poo [3] and Zhang et al. [4] performed more comprehensive experiments on cultures of rat hippocampal and xenopus tectal neurons respectively. They showed that the synaptic weight was potentiated if the pre-synaptic spike fired first. The smaller the time delay between pre- and post-synaptic spike, the greater the weight change. Again the reverse was observed if the post-synaptic spike fired first. The time window in which potentiation and depression occurred was $t_{post} - t_{pre} = \pm 20ms$. The shape of the weight change is shown in

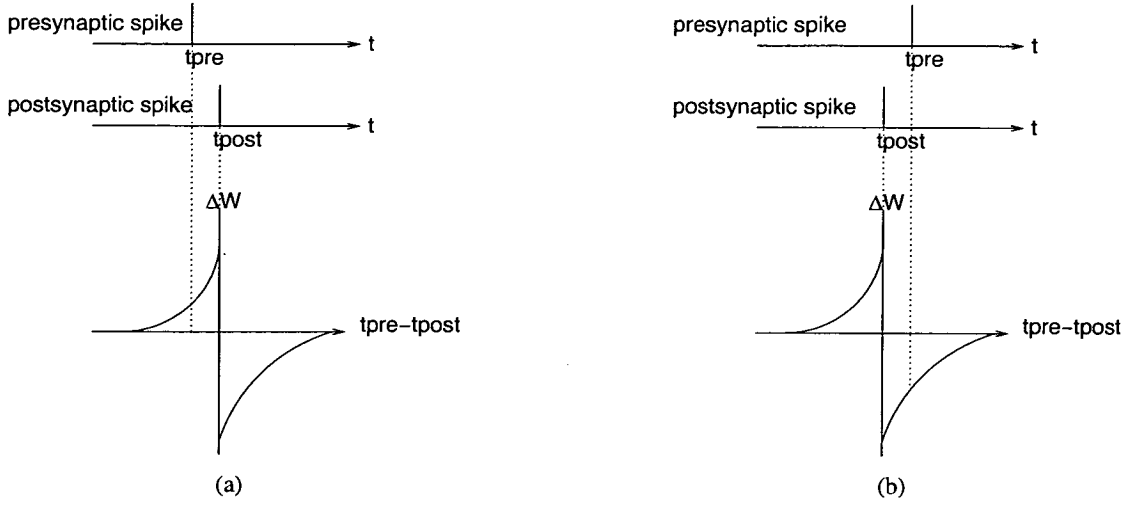


Figure 2.1: Temporally Asymmetric Hebbian Learning algorithm. The time between the post-synaptic firing time, t_{post} , and the pre-synaptic spike arrival time, t_{pre} , controls the amount of synaptic weight change, ΔW . The weight change curve is drawn using the assumption that the integral of the weight change curve is negative. This is necessary to keep learning stable if weight-independent weight change is implemented [44]. (a) illustrates the situation when the pre-synaptic spike arrives before the post-synaptic spike whereas (b) shows post-synaptic spike firing first.

figure 2.1. This particular type of spike timing dependent plasticity is referred to as Temporally Asymmetric Hebbian (TAH) learning from this point on.

Other biological studies have shown the spike timing dependent nature of synaptic potentiation and depression. The results have shown anti-Hebbian learning, where depression is induced when the pre-synaptic spike arrives first [45]. When the post-synaptic spike arrived first no significant potentiation was observed. Symmetric anti-Hebbian learning was observed in spiny stellate neurons [46] where the synapse was depressed if the time difference between pre- and post-synaptic firing was $\pm 10\text{ms}$. Asymmetric Hebbian rules have been found where the time difference over which depression occurs is considerably longer than that for potentiation [47, 48]. [49] and [50] provide reviews of the biological findings and mechanisms.

These spike timing dependent learning rules have been shown capable of having cooperative, associative and competitive behaviour. In Zhang et al. [4] they showed that if two synaptic inputs, each unable to trigger a post-synaptic spike themselves, fire at the same time the combined input can result in a spike which in turn will potentiate the two input synapses. Another experiment fired a weak synapse shortly before a strong synapse, causing it to be potentiated by association. In addition if both synapses are strong they will compete for the post-synaptic

spike. Simulations have shown that the definition of the weight change algorithm will have an effect on the final weight distribution. If weight-independent weight change is implemented, competition will dominate and the resulting distribution will tend to be bimodal [44]. In this case, stable weight change requires the integral of the weight change curve to be negative. With this condition, correlated inputs are more competitive and are potentiated while uncorrelated ones are depressed. Experiments performed by Bi and Poo [3] show weight-dependent weight change. Stronger synapses were potentiated less than weaker ones. Depression was weight-independent. This is a less competitive algorithm, and results in a unimodal distribution [51]. If correlated inputs are present, the mean of the distribution shifts to a higher level.

On a larger scale, networks have been shown capable of sequence learning [52, 53]. Boettiger and Doupe [54] reported experiments on the zebra finch forebrain which demonstrated that song learning may be one of the pattern learning tasks that uses spike timing dependent algorithms. Song and Abbott [55] performed simulations which used STDP to implement cortical re-mapping which may play a part in the brain's ability to recover from brain lesions. Spike timing dependent algorithms are also able to increase synchronisation within a network [5, 56].

Synchronisation can be achieved by adapting synaptic delays [6–8, 57, 58]. In [57] Gerstner et al. modelled the coincidence detection of the laminar nucleus of the Barn Owl [38] which is necessary for interaural time difference (ITD) detection. The variation in delay from the magnocellular neurons, which exhibit phase locking [37], to the laminar nucleus [38] was also modelled. The variation in delay will effect the phase information coming from the magnocellular neurons and prevent the ITD being calculated at the laminar nucleus. Gerstner et al. presented a spike timing dependent learning rule which could tune the delays, and therefore preserve the timing information. The phase locked input from the magnocellular neurons was modelled as occurring around a preferred phase of the stimulus tone, and included a $40\mu\text{s}$ temporal jitter. First the input was applied as a monaural input. After delay tuning phase locking to a 2kHz and 5kHz signal could be obtained with a precision of $20\mu\text{s}$ and $25\mu\text{s}$ respectively. This not only rejected those delays which degraded the performance, but improved on the temporal jitter at the input. The input path was then split into two in order to represent binaural input and the interaural time difference was added to one side. During learning the delays were adapted so that phase locking was maintained even with the ITDs incorporated. After learning the laminar neurons are sensitive to the particular ITD present during learning. This allows sound to be localised. The algorithm is powerful enough to achieve a temporal precision smaller than

the rise time of a post-synaptic potential. It is this ability which is particularly important to this work as transistor mismatch will affect the time constants of the design.

The circuitry required to assess whether a spike timing dependent algorithm can minimise the effect of transistor mismatch will consist of neuromorphic circuits. The following section introduces neuromorphic circuitry and describes adaptive circuitry, including STDP circuits, capable of learning and improving temporal correlations.

2.4 Neuromorphic Circuitry

In the late 1980s Carver Mead, a professor at the California Institute of technology, coined the term “neuromorphic electronics” to mean electronics where the principles of design were inspired from the biological nervous system. He argued that as the brain was far more computationally efficient than man-made methods of computation in terms of energy consumption, while being far better at interacting with the environment, circuit performance could be improved by learning from biology [59, 60].

Since then circuitry designed with biological principles in mind has been used both to advance knowledge about how the brain may work, and to create robust methods of solving real world problems.

There are many custom made circuits used in neuromorphic designs but most systems contain silicon neurons, synapses, which may or may not be adaptive, and a method for connecting them.

2.4.1 Silicon Neurons

In his 1989 book, [59], Carver Mead described a circuit which implements the Axon Hillock of a neuron. It is an integrate-and-fire neuron in that its input current is integrated on the membrane capacitor until the membrane potential reaches a firing threshold. The neuron spike causes the membrane, and therefore the neuron, to be reset. It is a simple circuit which only allows control over the pulse width. The membrane threshold is set at fabrication time. Therefore although it was used in [61], it has been superseded by circuits that allow additional control of circuit parameters such as membrane threshold and refractory period [9, 62, 63]. Neurons have also been designed that implement spike frequency adaptation [10]. This reduces the output firing

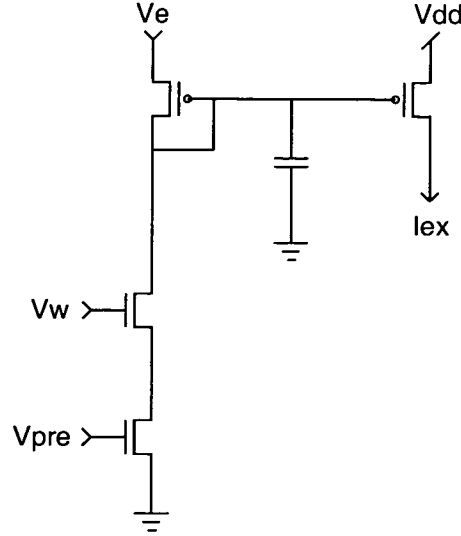


Figure 2.2: *Current mirror integrator configured as an excitatory synapse. After a spike is applied at V_{pre} an exponentially decaying current, with a time constant set by V_e , is supplied at lex . The weight of the synapse is controlled by V_w .*

rate if a large constant current input is applied. Another disadvantage to the neuron described in [59] is its high power consumption. A neuron optimised to reduce power consumption is presented in [64].

The integrate-and-fire model [27] is a simulation model which circuits emulate. When compared to biological neurons, it is considerably simpler but has been shown suitable for predicting the spike train of pyramidal neurons [65]. It does not, however, accurately represent the biological mechanisms that occur during spiking. The Hodgkin-Huxley model [66] is more complete as it includes information about individual ion channels within a neuron. Even though this model is more complicated in 1991 Mahowald and Douglas [67] reported a silicon neuron design based upon it. Hodgkin-Huxley type silicon neurons have been used in systems developed as tools for neurophysiology experiments [68].

2.4.2 Synapses

Synapses are used to provide the current input to the neurons. Usually the amount of current is controlled by a weight. These synapses can be excitatory, inhibitory or shunting inhibitory. These increase membrane potential, decrease the membrane potential and draw the membrane potential towards the resting potential of the neuron respectively. Excitatory inputs make it more likely a neuron will fire while inhibitory inputs make it more difficult.

One common configuration of synapse, which can be either excitatory or inhibitory is the current mirror integrator [62, 69], see figure 2.2 (excitatory configuration). After a spike it provides an exponentially decaying current with a time constant set by V_e when the circuit is biased sub-threshold. The weight is controlled by V_w . A mathematical description can be found in [69] and results from a fabricated circuit are reported in [62]. Alternative synapse configurations can be found in [9, 63, 70–72]. The circuit model in [72] models the short-term synaptic dynamics described in [73] and [74]. If a synapse is activated at a fast rate the magnitude of the synaptic output decreases. Therefore the firing rate of the associated neuron will decrease, and the resulting behaviour will be similar to that of the neuron with spike frequency adaptation [10].

One critical issue for synapses is the storage of the weight. Ideally it should be stable over a long time scale, but easily updated to allow learning. The easiest method for weight storage is across a capacitor. This is a volatile method of storage, and for any learning to remain in the absence of input activity, the weight voltage has to be refreshed. Ultimately this will mean quantising the weight voltages. In [75] the weight voltage is compared to a threshold and driven to either the maximum or minimum weight, resulting in a long-term bimodal distribution. A bimodal distribution may be all that is required for an application, but if a more continuous distribution of weights is required, the circuitry described in [76] drives the weight towards a range of attractor points, allowing a multi-state weight. Alternatively a wide range of weights could be used, if the quantising and refreshing was done using an analogue to digital converter, with the resulting digital weights stored in a digital memory. The analogue weight voltages could then be recreated when needed, by a digital to analogue converter. Unfortunately this approach requires a large area to be available for synaptic storage.

The approach in [70] and [71] is different. They use floating-gate transistors as the synapses. The weight can be stored in a non-volatile manner on the floating gate. While this solves the long-term weight storage problem, the characteristics of the programming mechanisms required for weight updates vary even across a single chip, requiring feedback controlled programming mechanisms to be used [77].

The weight, w , of a synapse can be defined as [78]

$$w = npq \tag{2.1}$$

where n is the number of release sites, p is the probability of synaptic release per site and q is

a measure of the effect of the post-synaptic effect of the synapse. This led to a further different approach being taken in [79] where, instead of varying the weight by adjusting q , the weight was quantised and then applied as a series of input spikes, e.g. a weight of four corresponded to four input spikes, varying n . In [80] the probability of release, p , could be controlled. This was achieved by gating the inputs spikes with a probabilistic signal determined by the weight; the stronger the weight the more likely transmission was.

Which method is used will ultimately depend on the system in which the weights are being used, the available space, the required length of storage etc.

2.4.3 Connectivity

When small networks are used to prove particular algorithms or circuit architectures, the internal connectivity can be easily managed and the outputs connected to output pins on the chip. As the circuits scale, in some cases containing over 1000 neurons, for example [81], a new method for interconnecting these neurons both internally and to the outside world is needed.

The method most commonly used is Address Event Representation (AER) [82]. It uses an asynchronous bus system to read out spiking events. When a neuron fires, it requests use of the bus, and its location is placed on it. The time of the spike is not needed as it is essentially being transmitted in real time. This address can then be used to index a look-up-table of connectivity, and spikes can be sent back in to all the synapses connected to it. The bus works considerably faster than the time constant of a neuron, allowing many events to be transmitted within the timescale of one neural event. An arbitration scheme has been designed which queues events if multiple neurons request access to the bus at the same time. While this will affect the timing of some of the spikes, it should be negligible compared to the neural timescale.

This method has been successfully used in many neuromorphic circuits with [83] describing a neuromorphic system consisting of five neuromorphic AER chips.

2.4.4 Applications and Learning Circuitry

Using circuits such as these building blocks, circuits have been designed that can perform various functions. The structure of the ear has been modelled resulting in silicon cochleas [84, 85] and low power cochlea implants [86]. Phase locking has also been implemented in

aVLSI. In [87] the authors presented a silicon neuron which modelled the bushy cell. The phase locking was measured using vector strength¹ and improved from 0.82 to 0.95 and from 0.6 to 0.97 for a tone of intensity 70dB and 110dB respectively. Silicon retinas have been designed which range from ones that directly reflect the structure of the retina [88] to transient image detectors that have spiking output [89]. The transient sensor described in [90] has been used as the input to orientation selective circuitry [91], and also a selective attention system [92] which uses a winner-take-all circuit to identify the important part of its field of view and orients the image sensor so that it is pointed at it.

Qi et al. [93] and Häfliger and Aasebø [94] both implemented time-to-first spike circuitry. In [93] a time-to-first spike CMOS imager was presented. The photocurrent from each pixel was integrated on the membrane capacitor of a neuron. The first spike was read out using an AER bus and then the pixel was disabled until the next frame. The time of the spike was then used as a measure of intensity allowing the image to be reconstructed.

Häfliger and Aasebø [94] implemented a rank order coder for 31 inputs. The input spikes were applied as AER pulses and then reconstructed using a current mirror integrator synapse and an integrate-and-fire neuron. The synapse that receives the highest number of spikes will have the highest output current and therefore the neuron associated with it will fire first. The spikes from the 31 neurons were connected to columns of winner-take-all circuits. The winner of the first column is the first spike to fire. All the other spikes are then passed to the next column which was won by the second fastest spike and so on. The circuitry was tested with ranked input patterns and the rank coder produced a good approximation to the input. The authors envisioned the circuitry being used with a silicon cochlea within a speech recognition system or in a vision processing system similar to that in [35].

Many of the learning rules used in neural simulations are unsuitable for aVLSI implementation. However, some spike timing dependent learning algorithms have been implemented due to their relative simplicity. Häfliger et al. [95] implemented a learning rule based on the Riccati equation [96]. It was capable of weight normalisation, weight-dependent weight change, and was shown to detect temporal correlations of 40%.

Floating-gate synapses have also been used to implement spike timing dependent learning rules

¹Vector strength is a normalised sum of unit vectors, one for each spike. A strength of one indicates perfect locking whereas a random phase distribution would have a strength of zero. [87]

[71, 97]. The nature of floating-gate programming results in these learning rules implementing weight-dependent weight change algorithms.

Temporally asymmetric Hebbian learning has been investigated [9, 10, 75, 81, 98–100] using a VLSI implementations. An alternative approach is described in [79]. The synaptic weight was held in the AER look-up table. Weight modifications were then implemented by reconfiguring the architecture of the associated AER circuitry to calculate the time intervals between spikes.

The circuits in [9] and [99] allowed full control of the learning algorithm parameters. The maximum and minimum weight change, width of window and area of potentiation and depression were all adjustable. The learning could be weight-independent or weight-dependent. A two layer network was fabricated. The first layer consisted of four neurons each with six synaptic inputs. The second layer neuron also had six inputs. The output of the four first layer neurons and two others.

In [99] it was shown that with the weight change set to be weight-independent, a single neuron could detect firing synchrony between its inputs. Each of the six inputs had a Poisson distributed spike train with a firing rate of 30Hz. Two of the inputs shared 20% of their spikes and the weight of their synapses were potentiated. Synchrony detection was also tested across the two layer network. Two of the inputs for each neuron shared 50% of their spikes. In addition 25% of the spikes in inputs 1 and 2 for neuron 1 were in inputs 1 and 2 to neuron 2. The 25% synchrony was detected at the second layer.

A test was also performed on the circuitry with weight-dependent weight change activated. In this case a single neuron could detect and potentiate the synapse with 10% common firings.

A further test with the weight-dependent learning rule was reported in [9]. In this test the spikes trains were correlated but did not contain synchronous spikes. Synaptic inputs 1 and 2 for the first layer neurons had correlation levels of approximately 5% and the second layer correlation level was lower. During learning the correlations were detected and the weights of the synapses with correlated input were potentiated. It was also shown that the learning process amplified the correlations between input 1 of neuron 1, and both the output of neuron 1 and the second layer neuron.

The circuitry designed by Arthur and Boahen [81] demonstrated the spike synchronisation property of STDP within a phase coding environment. Each neuron was inhibited by a leak-

age current modulated by an 8.3Hz theta rhythm. Variability in the process will cause some neurons to be more excitable than others, and therefore fire at an earlier phase compared to the theta rhythm. Applying STDP tended to cause strong excitatory connections to develop between neurons that responded quickly to the input stimulus, and ones that responded slowly. This in turn caused the slower neurons to fire faster, improving synchrony. If during learning the stimulus presented was a pattern, this pattern was learned. If a portion of the pattern was subsequently stimulated, the strong excitatory connections between neurons caused additional neurons to fire resulting in pattern completion.

The variability of the neurons reported in [81] is caused by transistor mismatch. Section 2.5 introduces transistor mismatch and describes some of the methods that can be used to reduce its effect.

2.5 Transistor Mismatch

Transistor mismatch is the term given to the variation in drain current in nominally identical transistors. In 1989 Pelgrom et al. [11] published a highly influential paper in which the mismatch in transistor threshold voltage (V_{T0}) and current factor (β) were derived.

$$\sigma^2(V_{T0}) = \frac{A_{V_{T0}}^2}{WL} \quad (2.2)$$

$$\begin{aligned} \frac{\sigma^2(\beta)}{\beta^2} &= \frac{A_W^2}{W^2L} + \frac{A_L^2}{WL^2} + \frac{A_\mu^2}{WL} + \frac{A_{C_{ox}}^2}{WL} \\ &\approx \frac{A_\beta^2}{WL} \end{aligned} \quad (2.3)$$

σ^2	variance
V_{T0}	threshold voltage with bulk-source voltage equal to 0
W	transistor width
L	transistor length
β	transconductance parameter, $C_{ox}\mu\frac{W}{L}$
C_{ox}	gate oxide capacitance per unit area

A_W	process related constant for W
A_L	process related constant for L
A_μ	process related constant for μ
A_{Cox}	process related constant for C_{ox}

The approximation in equation 2.3 holds true if W and L are large enough. Other factors become important if the spacing between devices is large or V_{sb} does not equal zero.

The resulting mismatch in drain current is then [11, 101]

$$\frac{\sigma^2(I_d)}{I_d^2} = \sigma^2(V_{T0}) \left(\frac{g_m}{I_d} \right)^2 + \frac{\sigma^2(\beta)}{\beta^2}. \quad (2.4)$$

I_d	transistor drain current
g_m	small-signal transconductance, $\frac{\partial i_d}{\partial v_{gs}}$, evaluated at the quiescent point
v_{gs}	transistor gate-source voltage

From equations 2.2 and 2.3 it can be seen that as device sizes reduce, mismatch will become more of a problem. It is somewhat mitigated by the fact that with each technology scaling A_{VT0}^2 reduces. This has been linked to a reduction in the oxide thickness (T_{OX}) [11, 102, 103] but as other factors change with technology scalings this can not be proved conclusively [104]. Unfortunately reductions in the power supply accompanying a technology scaling mean any mismatch in V_{T0} has a larger affect. The area of a minimum size transistor reduces with the square of the feature size. As reduction in A_{VT0}^2 is approximately linear with minimum transistor length, the overall effect is that the mismatch of a minimum size transistor increases with each scaling [13].

The equations in [11] were developed using a $2.5\mu\text{m}$ process. Since then further studies on device mismatch in smaller technologies have been undertaken which still show the $1/WL$ relationship to hold, except when the transistors are narrow and long or wide and short [105, 106]. As transistor dimensions become smaller the difference between WL and $W_{eff}L_{eff}$ becomes significant. Therefore

$$\sigma^2(V_{T0}) = \frac{A_{VT0}^2}{W_{eff}L_{eff}} \quad (2.5)$$

is a more accurate measure of V_{T0} mismatch [105]² where $W_{eff} = W_{drawn} - DW$ and $L_{eff} = L_{drawn} - DL$. It was also shown that as the length reduction factor (DL) was greater than the width reduction factor (DW), improved matching could be achieved with transistors of a smaller area if the drawn length was longer.

As the minimum feature size moves into the deep-sub-micron range, an additional mismatch term becomes important - the gate leakage mismatch [107]. This mismatch' contribution decreases with increasing W , but increases with L . As a result there is a minimum level of mismatch attainable while maintaining the W/L ratio [107]. There will be an improvement if W alone is increased, but at the expense of power consumption.

In order to reduce power consumption, the gate voltage overdrive ($V_{gs} - V_T$) can be reduced but this increases the effect of V_{T0} mismatch. As V_{gs} reduces, the mismatch levels increase until subthreshold operation is reached at which point it plateaus [11, 12, 108].

The drain current in a subthreshold transistor is [109]

$$I_d = I_0 S \exp \frac{V_{GB}}{nU_t} \left(\exp \frac{-V_{SB}}{U_t} - \exp \frac{-V_{DB}}{U_t} \right) \quad (2.6)$$

where I_0 is the zero bias current, S is the W/L ratio, n is subthreshold slope coefficient, U_t is the thermal voltage kt/q and V_{GB} , V_{SB} and V_{DB} are the gate to substrate, source to substrate and drain to substrate voltages respectively.

$\frac{q_m}{I_d}$ increases as V_{gs} reduces from strong inversion to weak inversion, and therefore $\sigma^2(V_{T0}) \left(\frac{q_m}{I_d} \right)^2$ becomes the dominant part of equation 2.4. If $\frac{q_m}{I_d}$ is calculated for subthreshold operation from equation 2.6, it is a constant. The plateau effect results from $\frac{q_m}{I_d}$ being constant in subthreshold [12]. When subthreshold mismatch is characterised specifically, it follows the $1/WL$ law in most cases [12, 108].

In [110] guidelines for optimum matching were set out. They are listed in table 2.1. These rules are primarily layout guidelines but rules 1, 3 and 8 should be considered during the design phase. Rules 1 and 3 come about as DL, DW and the short channel effect all cause variation in the current output of transistors with different areas but the same W/L ratio. Variations in oxide thickness, which effects threshold voltage, die orientation and temperature all cause gradient

²Lakshmikummar et al. [101] also determined mismatch levels with respect to effective width and length but much of the intervening work concentrated on W_{drawn} and L_{drawn} .

1	Same structure
2	Same temperature
3	Same shape, same size
4	Minimum distance
5	Common-centroid geometries
6	Same orientation
7	Same surroundings
8	Non minimum size

Table 2.1: *Rules for optimum matching [110]*

effects [104]. These gradients give rise to rules 2, 4 and 5. Common-centroid geometries ensure that matched transistors are laid out such that the centre of gravity of the transistors is the same. Therefore the gradients should effect all transistors equally. Figure 2.3 shows two examples of common-centroid layout for two transistors. (b) is the preferred layout, as it is common-centroid in both the x and y axis through placement, rather than just by the symmetry of the device, but it is more difficult to layout. This is especially true when large transistors are required, as it is preferable that no single transistor in the layout is very large. The dummy transistors shown address rule 7. The polysilicon etch rate depends on the distance to the nearest placement of polysilicon. Therefore, to ensure that any variation to the gate dimensions is consistent for all transistors, strips of polysilicon should be placed an equal distance away from the outside edge.

Pavasovic et al. [108] presented evidence of edge and gradient effects and their affect on the matching of subthreshold biased transistors. The edge effects caused NMOS currents to drop to 70% of mean current while the PMOS current rose to 180%. There were visible oscillatory variations in current with a high spatial frequency, which they termed striations, and longer-range gradients. They showed that after removing these systematic effects, the $1/WL$ relationship held.

To minimise the effect of mismatch during the design phase, the obvious solution is to increase the active area and in [111] a high level of current matching was achieved through careful layout and device sizing based on equation 2.4. However, this strategy ignores the benefits of moving to a smaller technology, and may not be possible if the additional capacitance associated with the larger devices reduces the speed of operation too much [13]. The circuitry should also be designed such that it is biased in a region where mismatch will have the least effect. The

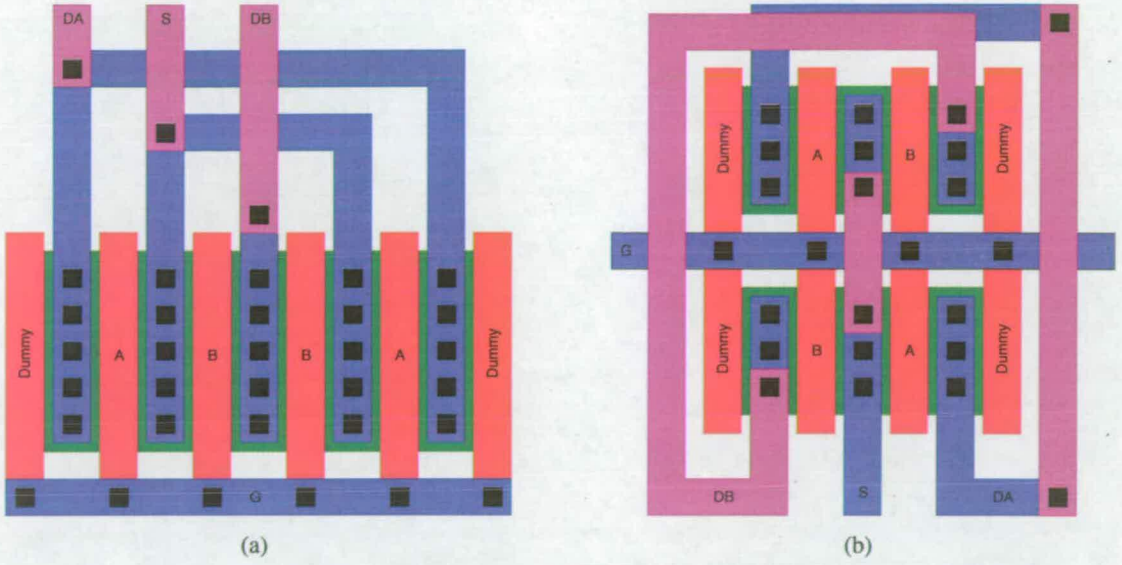


Figure 2.3: Common-Centroid Layouts of two transistors with common gate and source connections. Layouts of these types are used to improve matching by minimising the effect of gradients. (a) shows two interdigitated transistors. (b) shows two transistors laid out in a cross-coupled pair which is a two-dimensional common-centroid array.

gate voltage overdrive, $V_{gs} - V_T$, should be large when matched currents are required as any variation in V_{T0} has less effect. If identical currents are being applied and matched V_{gs} is the aim, the desired V_{gs} should be small [104]. It has also been reported that NMOS transistors match better [11, 101] and therefore should be used where possible. However in [106] results from a $0.4\mu\text{m}$ BiCMOS process showed that if the bias condition was set by a reference current, the PMOS transistors would match better. This comes about as for the same size of device and current, a PMOS will require a higher V_{gs} .

Post-fabrication trimming or calibration can also be used to improve the matching characteristics. Floating-gate devices were used in [17] and [18]. In [17] a current mirror was made using two floating-gate NMOS transistors. After programming, transistor mismatch for currents greater than $45\mu\text{m}$ was reduced to less than 0.1%. In [18] trimming was used to improve matching while the circuit was biased using a low gate voltage overdrive. This voltage bias was in the temperature insensitive range for the current source, allowing good matching across temperature gradients and in changing environmental conditions.

Maunu et al. [112] described another method for current source calibration. A single transistor provided I_q which was approximately 80% of the desired I_{out} . The remainder could be made

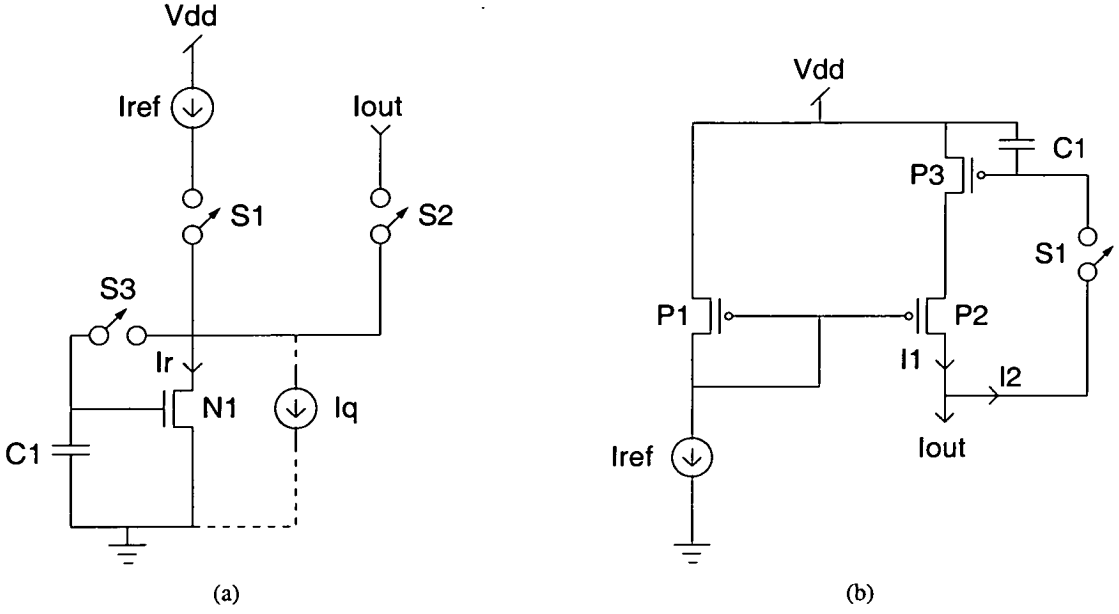


Figure 2.4: Dynamic Current Mirrors. (a) An example of the dynamic current mirror implemented in [113]. During calibration $S3$ and $S1$ are closed allowing a bias to be set across $C1$ so that $I_q + I_r = I_{ref}$. If I_q is not included the bias across $C1$ will be set such that $I_r = I_{ref}$ [14, 114]. During operation $S3$ and $S1$ are open and $S2$ is closed. I_{out} should now equal to I_{ref} . (b) An alternative calibration scheme where an offset to the source voltage of $P2$ is set by closing $S1$ while I_{out} is connected to I_{ref} .

up by switching in a selection of current sources, adding to I_q , until I_{out} fell within the desired accuracy. The smaller current sources were nominally $0.4I_{out}/N$ where N is the number of current sources. The finer current sources could have a larger standard deviation about I_q and were therefore smaller. Through simulation the authors showed that a mean current of $10\mu A$ could be matched with a standard deviation of 0.25% when 7 smaller current sources were used. Achieving this level of mismatch with a single transistor would require 3 times the area in the $0.13\mu m$ process simulated. The circuitry that compared I_{out} to the desired reference current was not shown and would have to be well matched and therefore large. This would decrease the area saving, but if a large number of calibrated currents were required and, importantly, if the simulations prove accurate, this could still be an area efficient method of ensuring matching.

In [113] a similar calibration technique, was performed but instead of selecting a set of smaller currents to make up the remainder, a dynamic biasing technique was used for the final 10% of the current, figure 2.4(a). I_q is nominally $0.9I_{ref}$ but will vary with mismatch. During calibration $S3$ and $S1$ are closed. Current is then sunk by I_q and $N1$ with the remainder causing the

voltage across $C1$ to rise until $I_q + I_r$ equals I_{ref} , $C1$ can be the parasitic gate-source capacitance, C_{gs} . During operation $S1$ and $S3$ are open and $S2$ is closed. I_{out} is now approximately I_{ref} , assuming the voltage across $N1$ and I_q is similar during operation and calibration and that any charge injection from the switching of $S3$ is small. Charge will leak off $C1$ over time which will affect I_{out} . In order to maintain a high level of matching, the calibration will have to be repeated at an interval determined by among other things, $C1$ and the g_m of transistor $N1$ [113]. When an array of these current sources was measured, the currents matched to within 0.02%

Others have implemented dynamic current mirrors, as the circuit in figure 2.4(a) is called, that do not have I_q in parallel with $N1$ [14, 114]. In this case the bias voltage across $C1$ is large enough that I_r equals I_{ref} . I_q was added in [113] to allow $N1$ to be sized in order to minimise the effect of charge injection during switching. It also had the effect that C_{gs} increased and the g_m of transistor $N1$ decreased, which in turn increased the time interval required between calibrations. However this method also has disadvantages. The current I_q must be less than I_{ref} . Therefore the transistors making up I_q will have to be large enough that mismatch does not cause the current to rise above I_{ref} . In fact it has to be kept low enough so that the bias voltage required for $N1$ is sufficiently large that I_q remains in saturation during calibration. Wegmann and Vittoz [14] provided their own method of minimising charge injection from switching. They designed bias circuitry which only turned on the switch transistor just enough to allow the bias current to be set. This reduced the amount of channel charge in the switching transistor and the charge injection caused by it. An accuracy of 0.05% was achieved.

Figure 2.4(b) shows an alternative dynamic matching technique described in [115]. In this case during calibration I_{out} is connected to a current source matching I_{ref} and $S1$ is closed. The difference between $I1$ and I_{ref} will then change the voltage across $C1$. This will affect the V_{ds} developed across transistor $P3$ which will change the V_{gs} of transistor $P2$ until I_{out} and I_{ref} match. During operation I_{ref} is disconnected from I_{out} and $S1$ is opened. I_{out} can only be reduced. Therefore this system can only correct I_{out} if it is greater than I_{ref} . A two-stage biasing scheme was introduced which allowed I_{ref} to be connected to I_{out} and allowed bidirectional compensation to be realised. In contrast to the previously described dynamic current mirrors, the output of this circuit is a matched current of the same polarity as the reference current. The results presented were from simulation but showed a matching ability of 1% for currents as small as 1nA.

Dynamic techniques have also been used to implement current dividers [15] and comparators [16]. The current divider uses a convergent algorithm to divide the input current by any integer N . The comparator uses dynamic biasing to remove mismatch within a differential pair.

Other techniques available for minimising mismatch effects beyond gate overdrive and increased area are ultimately going to be required [103, 107] as design constraints restrict the dimensions or power consumption. The aim of this work is to investigate whether spike timing dependent adaptation has the capability to be one of these other techniques. Section 2.6 describes the algorithm that was used as the context for this investigation.

2.6 Depth-from-Motion Algorithm

In 1999 Wörgötter et al. [21] proposed an algorithm that calculated the depth of an object in a scene, from the speed at which it passed over a radial flow field. This algorithm was chosen as the context in which to investigate the ability of spike timing dependent adaptation to minimise the effect of process mismatch. Some background details of the algorithm are presented here.

When a three dimensional point is projected onto a two dimensional plane, the transformations

$$x' = \frac{f}{z}x \quad y' = \frac{f}{z}y \quad (2.7)$$

occur where x, y and z are the coordinates in three dimensional space, f is the focal length of the lens and x' and y' are the coordinates of the projection [116].

These relationships show that the position of any projected point is proportional to the focal length of the lens, and inversely proportional to depth. This relationship provides the necessary information to allow the depth-from-motion algorithm to work. If the projected point remains at the same x and y position but there is a change in the depth (z), from the focal plane moving forward, the projected coordinates, (x', y') , will change in response and the depth can be calculated.

The two equations in equation 2.7 are sufficient to calculate depth if the speed of forward motion is known, but if the coordinate system is changed from Cartesian to polar coordinates,

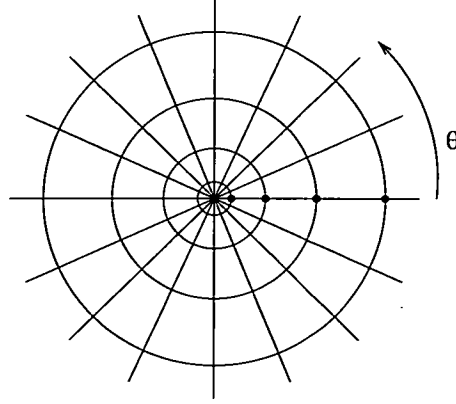


Figure 2.5: Layout of the artificial retina. *Neurons are positioned wherever the radial axis lines cross the concentric circles.*

the equations could be combined to give the single equation

$$r' = \frac{f}{z} r \quad (2.8)$$

where

$$r' = \sqrt{x'^2 + y'^2} \text{ and } r = \sqrt{x^2 + y^2}.$$

For this reason the retina was arranged radially, with a number of neurons along each axis as shown in figure 2.5. A radial flow field is induced along each axis by forward motion of the retina.

The neurons are designed to be excited whenever a sufficiently strong grey level change occurs. This effectively produces edge detection. The time difference between the excitation of two neighbouring neurons along a radii can be used to determine the depth of that edge. Full details can be found in [21].

Each neuron is only connected to its nearest neighbour on each side along the radius. For this algorithm it is assumed that the centre of the retina is positioned on the z-axis (depth), and that it is orthogonal to the direction of motion. This leads to a purely radial flow field as the retina is only moved along its optical axis.

The time difference between the firing of two neighbouring neurons is used. This means that only local operations are necessary and the algorithm can have a parallel implementation. While this is a computational advantage, it makes the algorithm highly sensitive to noise, for example

camera shake. To help reduce the effect of this noise, a prediction mechanism was added to estimate when the next neuronal excitation should take place. This is achieved by taking the position of the edge (calculated from the first two neuron firings) and a knowledge of the neuron placement, and using it to calculate a predicted time for the firing of the next neuron.

$$\Delta Z_p = r f \left(\frac{1}{r'_n} - \frac{1}{r'_{n+1}} \right) \quad (2.9)$$

ΔZ_p is the depth the focal plane must move forward, for an edge to move from neuron n to $n + 1$. r'_n is the position of the n th neuron on a radial axis of the retina. The time taken for the edge to move between neurons is then equal to ΔZ_p divided by the forward velocity.

The excitation of the third neuron will only be accepted as related to the first two if it occurs within a specified time window. When the third excitation occurs this can be used to refine the determined edge coordinates. If the depth is to be sampled at regular intervals the neuron placement should result in ΔZ_p being constant for all n . This was not the case in [21], but in a follow up paper [117], an attempt was made to do this. However it was found not to be possible as only a few neurons could be placed along a radius. Instead the retina was structured such that small sections sampled at regular intervals; but each section did not have the same sampling interval.

If this prediction mechanism is implemented in analogue VLSI, mismatch will inevitably degrade its performance. This work adds spike timing dependent adaptation circuitry to the prediction mechanism to minimise the effect of mismatch.

2.7 Summary and Conclusions

When Carver Mead introduced neuromorphic engineering in his 1989 book [59], many of the circuits operated in subthreshold where mismatch is a particular problem. Despite this he devoted very little space to resolving the problems of mismatch, but suggested that it was unwise to assume currents would be matched to anything closer than a factor of two, and that circuits should be designed that can tolerate this variation or perform self-compensation.

Spike Timing Dependent Plasticity is a neural algorithm, which operates on the timing between spikes, and has been shown to be capable of adapting synaptic delays. If the prediction element

of the depth-from-motion algorithm described by Wörgötter et al. [21] is implemented in analogue VLSI mismatch will compromise the timing of the output. It is the intention of this work to evaluate the ability of spike timing dependent adaptation to reduce the effect of transistor mismatch.

The following chapters describe the circuitry within this evaluation. Chapter 3 introduces the neuromorphic circuitry required to implement the prediction element of the depth-from-motion algorithm. Chapters 5 and 6 describe two implementations of spike timing dependent circuitry and report on their ability to minimise the effect of mismatch. The effect of mismatch on the system is also clearly visible.

Chapter 3

System Overview

3.1 Introduction

The algorithm described in section 2.6 has many interesting aspects. However, this work concentrated on the prediction element. This resulted in the system shown in figure 3.1. The spike firing circuit is made up of three integrate-and-fire neurons that are set to fire at predetermined intervals. For example, the delay between neurons one and two firing may be the same as the delay between neurons two and three. In a complete system the intervals would be determined by pixel spacing. The spikes from neurons 1 and 2 are passed to the prediction network which generates another spike, spike 3p (i.e. “predicted”, not actual), at a time determined by the firing times of the input spikes. The timing of this spike can then be compared to that of spike 3 by the spike confirmation block. Spike 3 will only be accepted as genuine if it arrives within a given time window of spike 3p. The adaptive STDP network provides feedback to the prediction network, determined by the difference between the actual firing time $t(3)$ and its predicted time $t(3p)$.

The following notation will be used: neuron x fires spike x at $t(x)$. If a spike name ends with p , e.g. spike xp , it is the prediction of the spike x .

3.2 Integrate-and-Fire Neuron

Figure 3.2 shows an analogue VLSI leaky integrate-and-fire neuron based on those shown in [62] and [118]. The input current, I_{in} , causes the voltage across C_m to rise. When this rises above V_{th} the neuron output goes high. C_{fb} applies positive feedback to reinforce the spike. The refractory circuitry also goes high, which discharges C_m at a rate controlled by V_{pw} . V_{bref} controls the length of the refractory period, i.e. the minimum time possible between spike firings. V_{leak} can be set to ensure any charge on C_m leaks away over time if the neuron does not fire. This is particularly useful if the neuron is being used to identify coincidence in inputs or integrating the output current from a pixel, as it can be used to subtract the dark current.

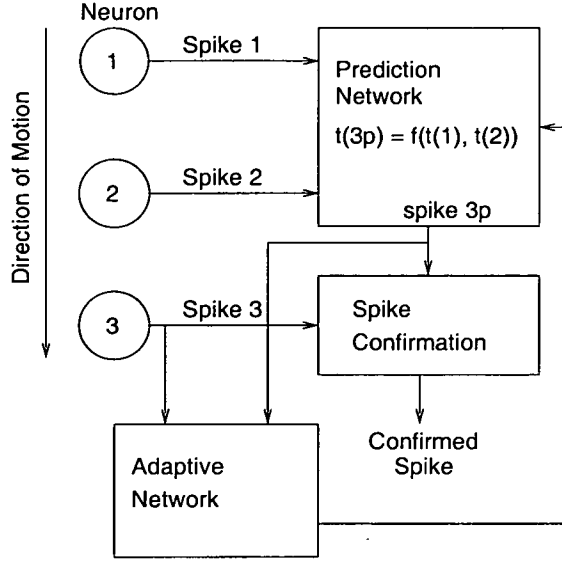


Figure 3.1: System Block Diagram. Neurons 1 to 3 fire at predetermined intervals. Neuron 3p fires at a time predicted from the firing times of neurons 1 and 2. If the neurons fire at equal intervals $t(3p) = t(2) + \{t(2) - t(1)\}$ where $t(1)$ is the firing time of spike 1, $t(2)$ for spike 2 etc. Spike 3p is the prediction of spike 3.

The component sizes and bias voltages are detailed in appendix B.

3.3 Confirmation Block

The confirmation block is shown in figure 3.3. The Neuron block is the same as that shown in figure 3.2 but V_{leak} is set in combination with V_{wfc} so that only two spikes arriving within a fixed time, e.g. $|t(3) - t(3p)| < 500\mu s$, will cause the neuron to fire. I_{in} from one spike is not enough to cause the neuron to fire. The leakage current causes the membrane voltage to drop, such that if the second spike arrival does not happen within a short enough time, the neuron will not fire. The refractory period of the integrate-and-fire neurons can be set long enough that two spikes from the same neuron cannot cause the confirmation spike to fire.

3.4 Prediction Network

The spacing of the pixels on the artificial retina determines the relative times between the firing of spikes 1, 2 and 3. The initial assumption is that the retina has been designed such that the time delay between spikes 1 and 2 equals that of spikes 2 and 3. Therefore the prediction is

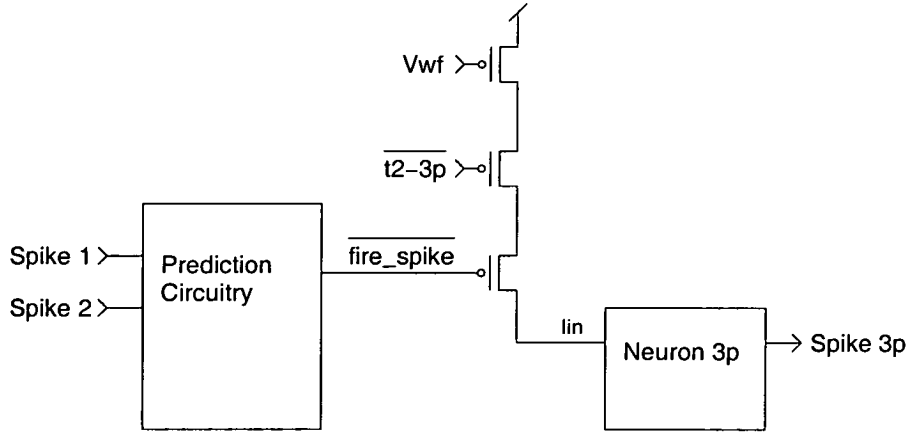


Figure 3.4: Prediction network. *The prediction circuitry initiates a signal when it predicts spike 3 will fire which causes neuron 3p to fire. This prediction is based on the timings of spikes 1 and 2.*

3.5 Adaptive Network

The adaptive network takes spikes 3 and 3p as its input. The circuitry uses the relative timings of these spikes to provide feedback to the prediction circuitry, allowing an improved prediction. The methods used are described in chapters 5 and 6.

3.6 Summary

This chapter has introduced the circuitry that will be used to implement the prediction element of a depth-from-motion algorithm. The circuitry for the integrate-and-fire neurons and confirmation block is described. The interactions they will have with the prediction and adaptive networks are shown. The full circuit details can be found in subsequent chapters.

Chapter 4

Prediction Circuit Architectures

4.1 Introduction

The prediction circuit has to be able to issue a signal to fire a spike at a time determined by the arrival of spikes 1 and 2 and the neuron placement on the retina, see section 3.4. In this implementation, it is assumed that the interval between spikes 1 and 2 is the same as that between spikes 2 and 3, but any design should be flexible enough to allow other ratios.

Three possible architectures are presented here. The first uses a single current source which charges two capacitors, the first charging between spikes 1 and 2 and the second between spikes 2 and 3p. When the voltage across the second capacitor reaches the voltage across the first, spike 3p can be fired. The other two are based on the principle of charging and discharging a single capacitor with matched currents. If the currents are perfectly matched the voltage across the capacitor will return to its starting level after the appropriate delay has occurred. A spike can then be fired.

4.2 Two Capacitor Circuit

This circuitry is based on the principle that capacitor matching is superior to that of currents. The circuitry is shown in figure 4.1. The control signals $\overline{t1 - 2}$ and $\overline{t2 - 3p}$ are generated by spike triggered SR Latches, e.g. set connected to spike 2 and reset connected to spike 1 results in $\overline{t1 - 2}$.

The sequence of events is as follows:

- t(1) Spike 1 is fired and C1 and C2 are discharged through N1 and N2 respectively.
- t(1)-t(2) P2 is on and C1 is charged by the current set by Vb .
- t(2) P2 turns off while P3 becomes active.
- t(2)-t(3p) C2 is charged by the current set by Vb .

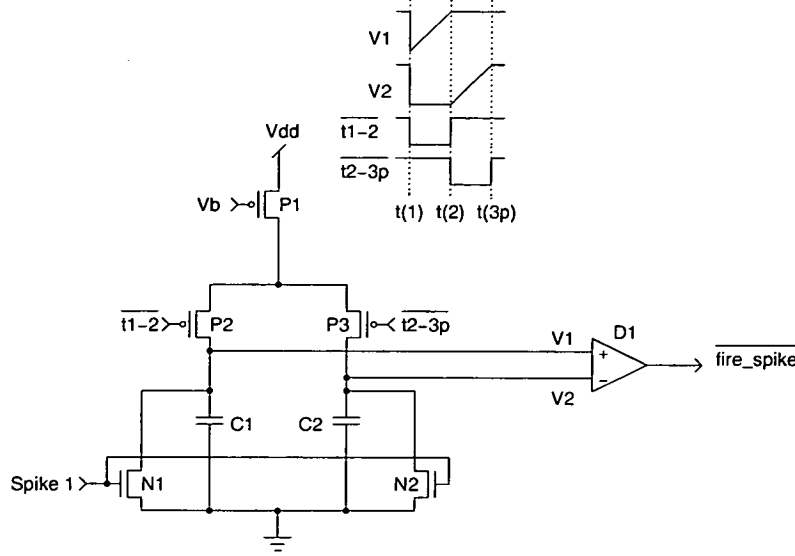


Figure 4.1: Prediction circuit using two capacitors. $C1$ charges between $t(1)$ and $t(2)$ and $C2$ charges from $t(2)$ to $t(3p)$. A spike is issued when $V2$ rises above $V1$.

$t(3p)$ When $V2$ rises above $V1$ the fire spike signal is activated ending the sequence.

In order to test the circuit's response to mismatch, the Cadence Monte Carlo simulation tool was used. This varies the SPICE transistor parameters, V_{T0} (threshold voltage) and $U0$ (mobility) by an amount related to the observed variance in a process, in this case the Austria Microsystems (AMS) CSI $0.35\mu\text{m}$ process. The circuit was subjected to 100 runs, the error in the prediction, $t(3p) - t(3)$ was measured, and the results can be seen in figure 4.2. The time between spikes 1 and 2 was 5ms.

The prediction error results are mostly within the $\pm 500\mu\text{s}$ range. The slight negative bias to the distribution is caused by the charge injection from $C1$ to $C2$ when $P2$ and $P3$ switch. The effect of charge injection is to reduce $V1$ while raising the starting point for $V2$. While the capacitors will be better matched, the accuracy of the prediction now relies on the ability to reduce charge injection from the switching of $P2$ and $P3$, and the matching within $D1$.

4.3 Dynamic Current Mirror

Another way to predict the firing of spike 3 is to charge a capacitor between $t(1)$ and $t(2)$, and then discharge it with a matched current source after $t(2)$. When the voltage across the capacitor returns to its starting point, spike 3p should be fired.

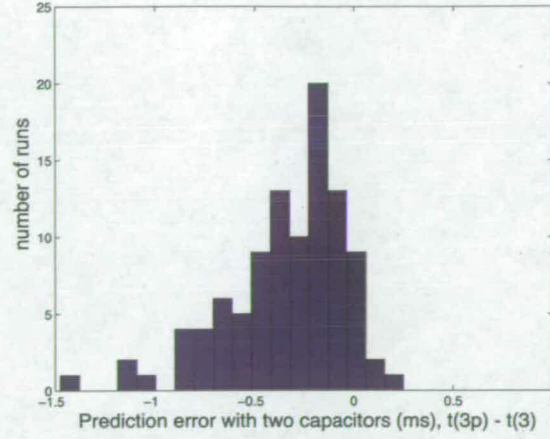


Figure 4.2: *Distribution of prediction errors for the two capacitor prediction circuitry. One hundred Monte Carlo simulations were run and the prediction error, $t(3p) - t(3)$, was measured. $t(2) - t(1) = 5ms$*

The significant source of error in this prediction will be any mismatch between the charging and discharging currents. The dynamic current mirror, described in [14], is an architecture designed specifically to minimise the effects of mismatch. The dynamic current mirror prediction circuit is shown in figure 4.3.

The sequence of events is as follows:

- t(1) Spike 1 is fired and P4 and N4 act as a switch which sets the initial value of V_{ramp} to V_{rampth} .
- t(1)-t(2) P2 is on and C1 is charged by the current set by V_b .
- t(2) P2 turns off while N2 becomes active.
- t(2)-t(3p) C1 is discharged through N1 and N2. The discharging current is set by the voltage across C2.
- t(3p) V_{ramp} is compared to V_{rampth} using a differential pair. When V_{ramp} crosses it, neuron 3p is “fired”. This ends the discharge period.
- t(3p)-t(1) N2 is now turned off and P3 becomes active. The bias current will now flow through N1 or through N3 charging C2. If the voltage across C2 is too small to allow the full current to pass through N1, the remainder flows through N3 and charges C2, therefore raising the amount of current able to pass through N1. When the voltage across C2 is high enough that all the current flows through N1, no further charging of the capacitor take place.

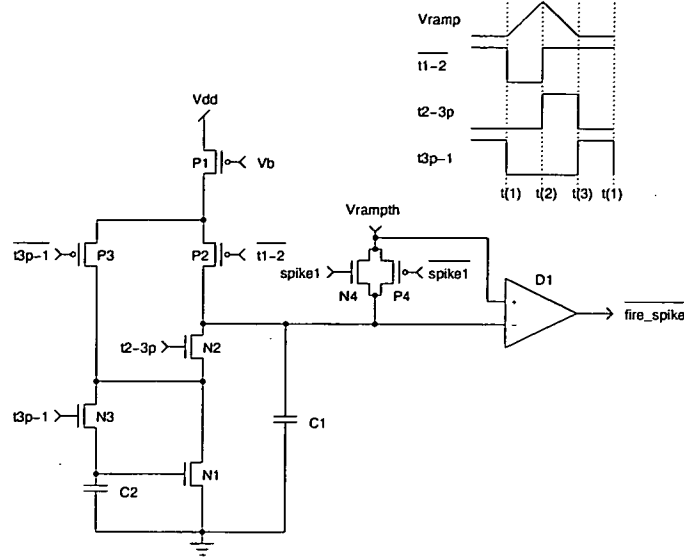


Figure 4.3: Dynamic current mirror circuit. *C1 charges between $t(1)$ and $t(2)$ and discharges from $t(2)$ to $t(3p)$. When V_{ramph} falls below $V_{ramphth}$ spike 3p is fired. Between $t(3p)$ and $t(1)$ C2 is charged to the voltage required to let all of the bias current pass through N1. This then sets the discharge current.*

Although this should provide perfect matching, when N3 switches off at $t(1)$ the switching causes charge injection which alters the bias voltage by an amount $\Delta V = \Delta q / C2$. The voltage across C1 can also couple to the voltage across C2 changing the bias voltage to N1 as V_{ramph} changes. The circuit was also tested using Monte Carlo simulation and the results can be seen in figure 4.4.

The results show that a dynamic current mirror can reduce the error to a similar range as the two capacitor circuit. The prediction has a positive bias as the switch induced charge injection tended to cause a reduction in the voltage across C2.

At this point it was important to establish what an acceptable prediction error was. In [21] a spike was accepted as accurate if it arrived within a finite number of time steps from the predicted spike arrival time. The authors were approached for further information as to how the window of acceptance compared to the average prediction time but no information was received. A decision was then made to use the circuits described in sections 4.2 and 4.3 to set the minimum level of accuracy required from the circuit in section 4.4 after adaptation.

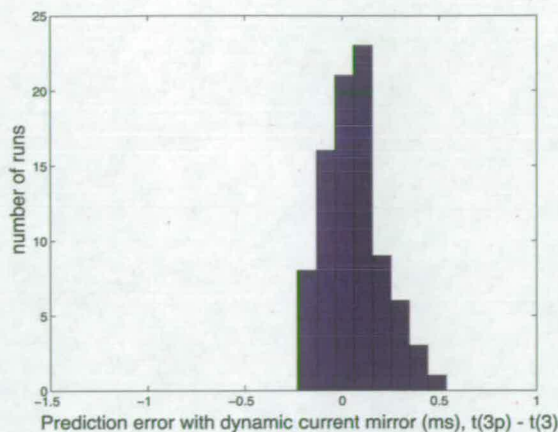


Figure 4.4: *Distribution of prediction errors for the dynamic current mirror prediction circuitry. One hundred simulations were run and the prediction error, $t(3p) - t(3)$, was measured. $t(2) - t(1) = 5ms$*

4.4 Circuitry suitable for spike timing dependent adaptation

The two previous circuits can provide a prediction but are not completely error free. However their performance can be used as a guide to the minimum acceptable performance from an adaptive circuit.

In order for adaptation to be applied, a prediction circuit had to be designed that would allow this. It is also preferable that the circuitry can be easily altered to provide a 1:x prediction rather than just restricting it to the initial assumption of 1:1. The circuitry chosen is shown in figure 4.5.

The architecture of this prediction circuit is similar to that of the dynamic current mirror but without the dynamic biasing. The sequence of events is as follows:

- t(1) Spike 1 is fired and P6 and N6 act as a switch which sets the initial value of V_{ramp} to V_{rampth} .
- t(1)-t(2) P5 is on and C1 is charged through P3 and P4 which supplies a mirrored version of the current set by V_b .
- t(2) P5 turns off while N5 becomes active.
- t(2)-t(3p) C1 is discharged through N3 and N4. If the current source and sink are matched the time taken to discharge C1 should match the charging time.
- t(3p) V_{ramp} is compared to V_{rampth} using a differential pair. When V_{ramp} crosses it,

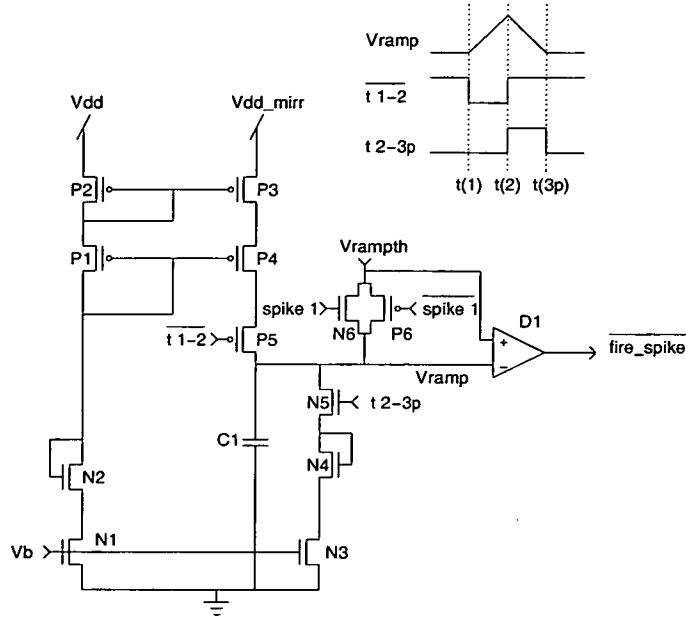


Figure 4.5: Spike prediction circuit. *C1 charges between $t(1)$ and $t(2)$ and discharges from $t(2)$ to $t(3p)$. The charging and discharging currents are nominally matched to enable prediction of spike 3 at the time when C1 has returned to V_{rampth} .*

neuron 3p is “fired”. This ends the discharge period.

The accuracy of the prediction depends upon how well the charging and discharging currents are matched. We are primarily interested in gain mismatch, i.e. differing $\frac{I_D}{V_{gs}}$ at the fixed dc bias point set by V_b (figure 4.5), and therefore transistors P1, P4, N2 and N4 are used to minimise the current change caused by the short channel effect. As this circuit has an intrinsically poor matching ability, very little information would be gained from a figure showing the prediction error over 100 Monte Carlo simulations. Instead, figure 4.6 shows the result of three different Monte Carlo simulations. Firstly the charging and discharging currents are similar, resulting in a prediction with an accuracy comparable to the other two circuits. In (b) the charging current is smaller than the discharging current, resulting in an early prediction. In (c) the situation is reversed and the prediction is late.

It should also be noticed that the addition of a second power supply pin, V_{dd_mirr} , to the circuit in figure 4.5 allows mismatch to be forced on to the circuit. This facilitates testing of the adaptive network under a wide range of induced prediction errors, which would not be possible with the limited number of chips received. Under most test conditions the two pins are shorted together.

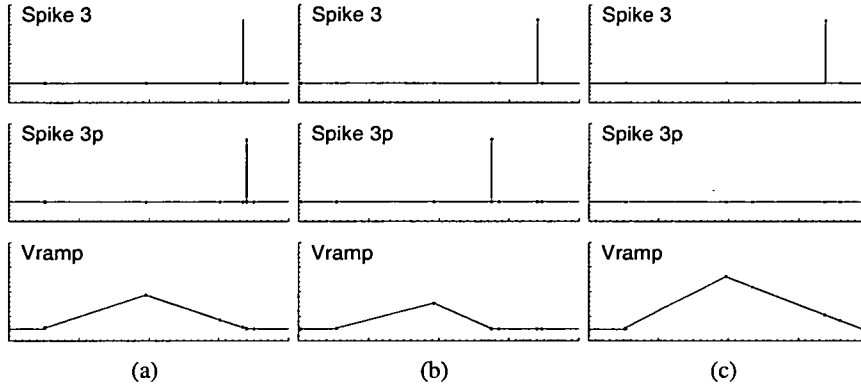


Figure 4.6: Pre-adaptation waveforms. These results show the spike 3, spike 3p and Vramp signals during three process mismatch Monte Carlo simulations. These waveforms show the possible effect of mismatch. First the charging and discharging currents are similar, resulting in a prediction with an accuracy comparable to the other two circuits. In (b) the charging current is smaller than the discharging current, resulting in an early prediction. In (c) the situation is reversed and the prediction is late.

The requirement that it should be possible to ratio the charging and discharging currents is easily met by altering the sizes of transistors N1 and N3 in figure 4.5.

4.5 Conclusions

Three possible prediction circuits have been described. The two capacitor approach and dynamic current mirror are more immune to the effects of mismatch than the adaptation suitable approach and provide a guide for the minimum effectiveness required from the adaptive network.

Comparing the size of the existing circuits the third circuit is smaller as it only contains one capacitor. It will of course be added to by the adaptive circuitry, which will offset this difference.

In [117] it was asserted that while it is desirable to have equal delays it is not possible at all times. This presents a problem for the dynamic current mirror circuitry. It is possible to produce multiples of the input current, but this requires more complicated switching circuitry. Either the dynamic biases would have to be refreshed during a fraction of $t(3p) - t(1)$ or, the refreshes would have to be take turns. This has implications for the size of capacitor required to store the bias. The other two methods could be adapted, with the transistors being sized in the adaptive method, and the capacitors in the two capacitor circuit. It is preferable to size

transistors than capacitors due to their relative size.

For that reason, and because other sources of mismatch can enter the circuitry, such as a neuron firing faster than others, track delay or the non-infinite gain of the differential pair, it was considered worthwhile to investigate the ability to adapt out the effects of mismatch using spike timing dependent adaptation.

Chapter 5

Spike Timing Dependent Adaptation

5.1 Introduction

To minimise the prediction error, a method must be found either to adjust one, or both, of the currents in figure 4.5 until they are matched, or to compensate for the effect of the difference in the currents. Changing the voltage across C1 at the time of the switching between charging and discharging, $t(2)$, will alter the discharge time. The voltage can be increased or decreased, which has a direct effect on the discharge time, and therefore compensates for the current difference. This method was chosen and examples are shown in figure 5.1(b).

5.2 Adaptive Circuitry

The change in the voltage across C1, figure 4.5, is achieved using the excitatory/inhibitory synapse described in [119], and is shown in figure 5.2 along with the output waveform. The circuit is designed such that if V_w , is greater than V_{synth} , the synapse is excitatory. Should V_w be smaller, the synapse is inhibitory. A differential amplifier could also be used to achieve this effect with a reduced number of transistors but to achieve the $\pm 10\mu A$ output current over a large input range the bias transistor would have to have a large W/L ratio. In addition, the contribution to knowledge in this work comes from the mismatch compensation circuitry and not the synapses. Therefore, rather than attempting to minimise the circuit, an existing design which met the requirements was used. *lin/out* is only connected to V_{ramp} at $t(2)$, and the value of V_w will determine the amount and the direction that the peak voltage will change. The width of spike 2, and therefore the time *lin/out* is connected to V_{ramp} , will be constant for a given set of environmental conditions. Should these change, V_w should change accordingly.

Transistors N1 and N6 are both small W/L transistors cascoded by the wide transistors N2 and N7. This maintains N1 and N6 in the linear region over a wide range as described in [98] and [118]. N1 is biased by V_{synth} into the linear region. The value of V_{cas} determines the minimum value of V_w at which N6 enters the linear region and will set the maximum amount of current

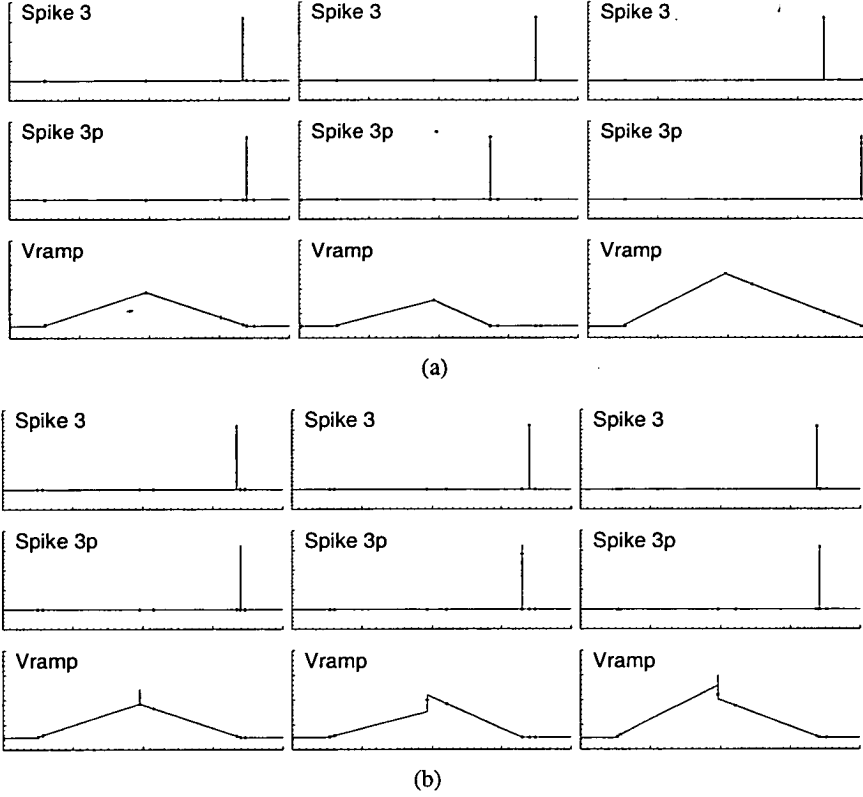


Figure 5.1: Pre-and post-adaptation results. These results show the spike 3, spike 3p and Vramp signals during three process mismatch Monte Carlo simulations. (a) is pre-adaptation and the error in spike 3p is clear. (b) shows the signals post-adaptation where the error has been reduced. The alteration to Vramp can also be seen.

that can be injected/removed from C1. It also determines the gradient of the I/V curve, which has a direct effect on the amount the output current changes with a step change in V_w . The smaller the range of required current the more accurate an output current can be selected.

Most variants of spike timing dependent plasticity maximise the synaptic weight change when spikes are near-simultaneous. In contrast, this system must be designed to apply weight changes which result in the spikes becoming more coincident. This method most closely relates to the experiments in [56].

The setting of V_w must be controlled, in order that the appropriate value is set to give coincidence of spikes 3 and 3p. The circuit shown in figure 5.3 provides this control by implementing the weight change curve shown in figure 5.4.

The circuit in figure 5.3 has two distinct parts. MOSFETs N1-3 and P1-3 control reductions in

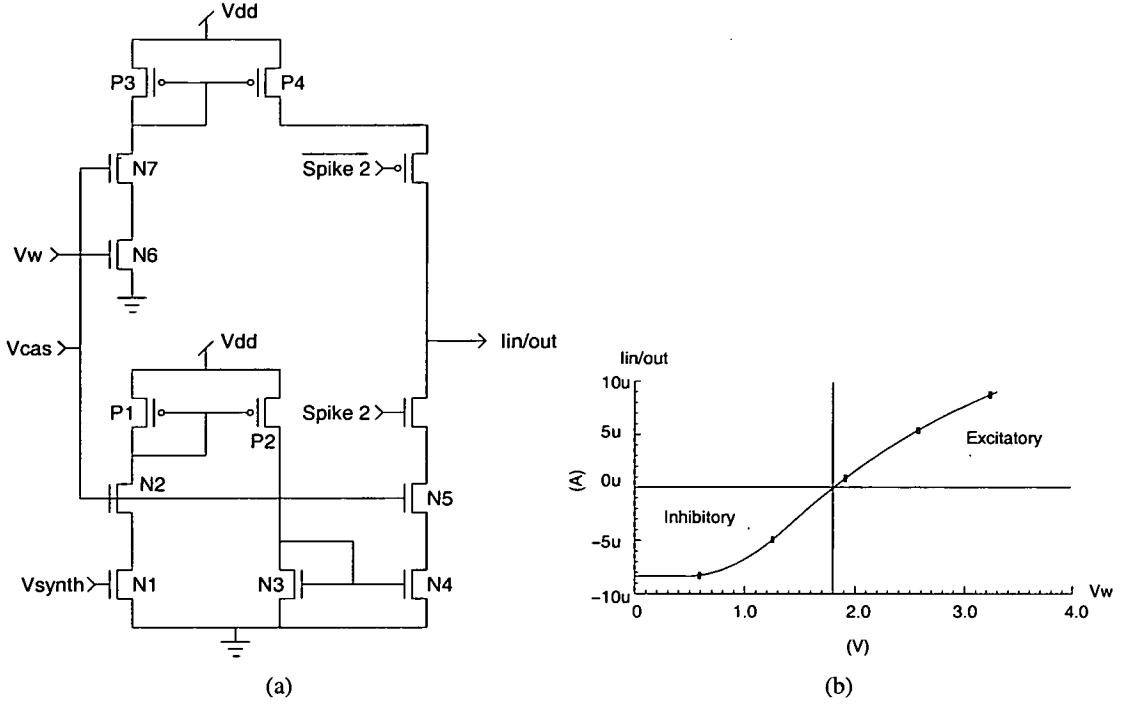


Figure 5.2: Excitatory/inhibitory synapse circuit and I/V graph. The synapse is excitatory if V_w is greater than V_{synth} and inhibitory if V_{synth} is greater. (a) shows the circuit configuration. (b) is the synapse output current with $V_{synth} = 1.8V$. $I_{in/out}$ is zero when $V_w = V_{synth}$.

V_w (depression) through N7-9, while increases (potentiation) are achieved through transistors N4-6 and P4-9. As the potentiation and depression circuits are mirror-images, with transistor polarities changed, a detailed description of only the depression mechanism is presented.

When spike 3 occurs C1 is discharged through N1 and then charges slowly to V_{dd} through P1. This results in a pulse at the gate of N8, whose width is determined by V_{b2} . This pulse defines the window within which weight change occurs. At $t(3)$ C2 is discharged through N3 and then charged slowly to V_n through P3. This voltage is N7's gate voltage and controls the size of the weight change. As the voltage across C2 increases with time, the longer the delay between $t(3)$ and $t(3p)$ the greater the weight change. At some point the voltage will reach V_n , this defines the maximum weight change. If spike 3p occurs within the time window, Cw is discharged through N7, N8 and N9 by an amount related to the voltage across C2. When the weight voltage is decreased, the peak of V_{ramp} decreases. This reduces the discharge time and spike 3p will occur at a time $t(3p)$ that is closer to $t(3)$.

The weight change is "capped" by V_n and V_p to prevent V_w moving from one supply rail to the

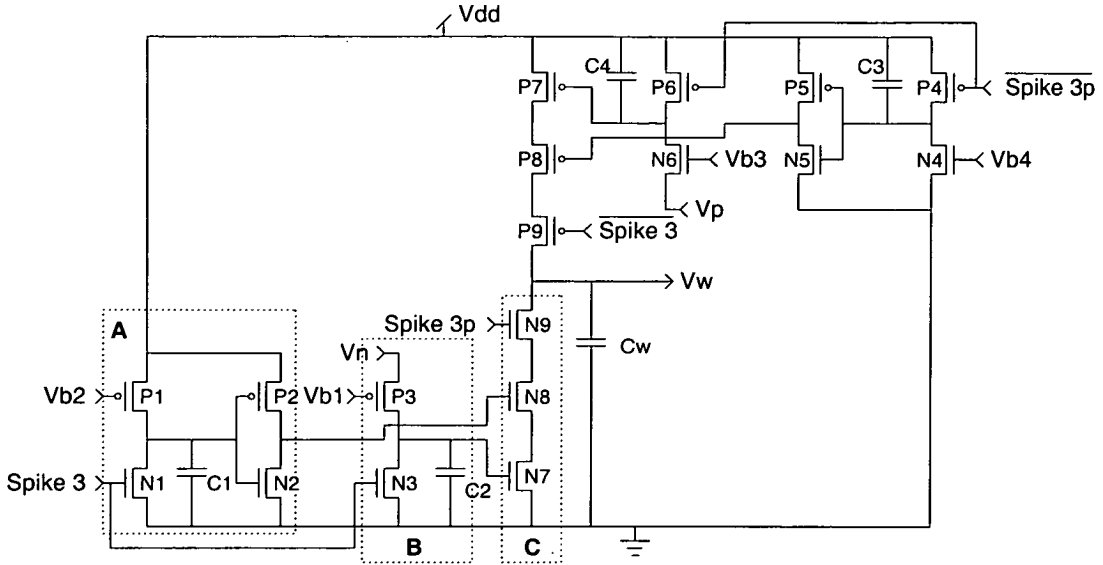


Figure 5.3: Weight adaptation circuit. V_w is increased if spike 3p occurs before spike 3 and decreased if neuron 3 fires first. Weight change only occurs if spikes 3 and 3p occur within a specified time window. The weight change (ΔW) is “capped” by V_n and V_p to prevent V_w moving from one supply rail to the other. Three circuit blocks are highlighted. **A:** The circuit setting the time window. **B:** The circuit setting the amount of weight change. **C:** Combining the two together to create the negative part of the ΔW graph.

other. The window width is determined empirically from *a priori* knowledge as to when two spikes may be regarded as unrelated events. In this case the window was 5ms.

There is a null point caused by charge injection due to switching around $t(3p) - t(3) = 0$, (see inset figure 5.4,) which determines how close spikes must be before the time difference has a negligible effect on the weight. When choosing the bias voltages, $Vb1$, $Vb3$, Vn and Vp , there is a trade off to be made between speed of convergence and the width of this null point, determined by $Vb1$ and $Vb3$. This results in a tolerance associated with the prediction. The maximum weight change set by Vn and Vp has an effect on how many spike pairs are required to correct a particular prediction.

If the adaptation windows set by $Vb2$ and $Vb4$ are too small, large mismatch errors will be judged to be uncorrelated spikes, and no adaptation will occur. This leads to a further trade off between the circuit’s ability to adapt and the maximum allowable density of edges in the scene. A larger window will allow a greater range of errors to be corrected but, for the circuitry to function, no two edges can occur within that interval.

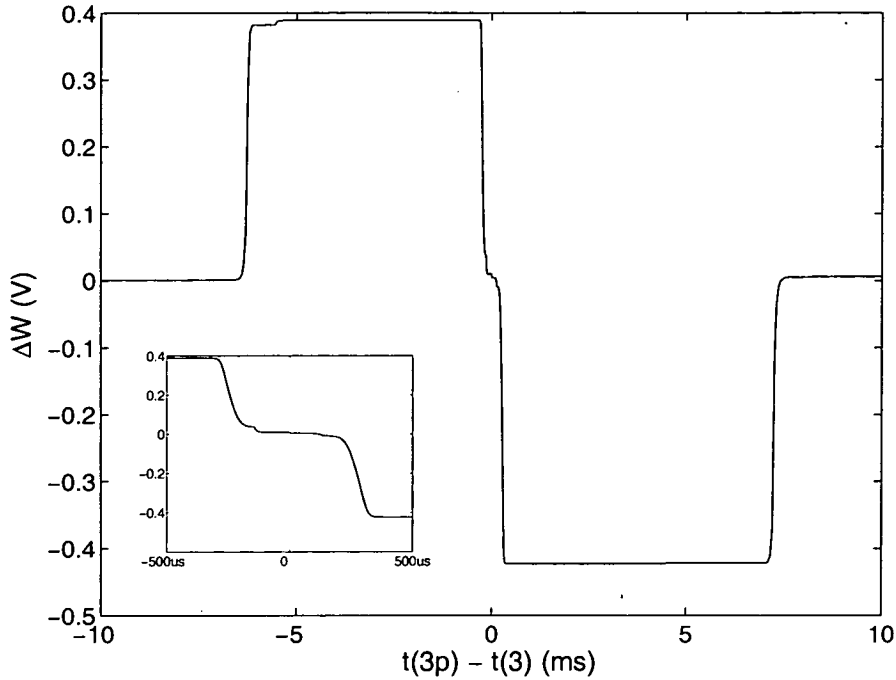


Figure 5.4: Weight adaptation algorithm. The weight change algorithm implemented by the circuitry shown in figure 5.3. The change in ΔW was measured in simulation. The inset picture is the range $-500\mu s - 500\mu s$. The null point around $t(3p) - t(3)$ occurs when charge injection due to switching dominates.

When spike 3p is a good prediction of spike 3, no further weight change occurs. It is assumed that correct edges will vastly outnumber incorrect edges, and that if one does fall within the adaptation time window it will only cause a small deviation of weight which will be corrected by subsequent correct spikes. The adaptation network can therefore be left active after this “self-calibration” and will continue to compensate for any circuit drift caused by changes in environmental conditions or charge leakage from the capacitor. The latter is particularly important as many compensation techniques suffer from problems in the permanent storage of calibration values.

5.3 Simulation Results

In order to establish whether the adaptive circuitry successfully improved performance when mismatch affects the system, the circuitry was subjected to a 100 run Monte Carlo simulation. It was required that the performance after adaptation be at least of the level achieved by the other prediction methods, see sections 4.2 and 4.3.

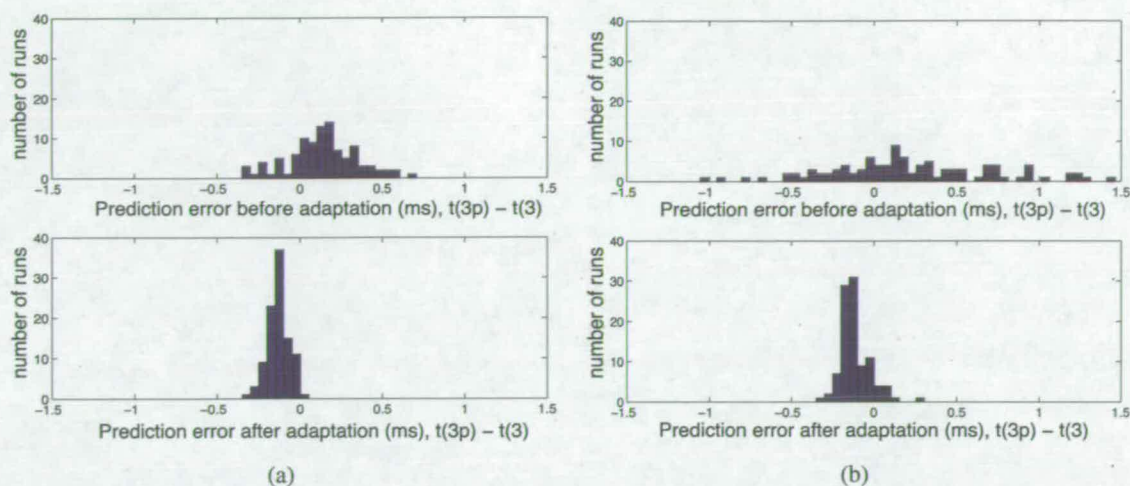


Figure 5.5: Distribution of prediction errors. The prediction error is defined as $t(3p) - t(3)$. Each graph has the results before adaptation at the top with after adaptation underneath. Results (a) were from a simulation when $t(2) - t(1) = 1\text{ms}$ and (b) $t(2) - t(1) = 5\text{ms}$

The circuits were simulated with an inter-spike interval of 1 and 5ms. The time between the prediction and actual spike was measured before and after adaptation for each run, and the results are shown in figure 5.5. The 1ms results are shown in (a) with the 5ms results in (b). If the post-adaptation results are compared to those in figures 4.2 and 4.4 the prediction error is of a similar level. The 1ms results show less of an improvement, but this is mainly due to the fact that the pre-adaptation error is approximately proportional to the time delay applied, and therefore the initial error is reduced.

As the simulation results showed potential it was decided to fabricate the circuitry.

5.4 Chip Architecture

If the full depth-from-motion algorithm was to be realised in aVLSI, a method to detect the transient edges would be needed. Transient edge detecting pixels were designed by Vasin Boonsobhak, another student working in the department, which provided enough current output to initiate a neural spike. Full details can be found in [120].

On the final chip layout, see figure 5.6, one block of prediction/adaptive circuitry with voltage activated inputs was laid out, along with two pixel activated blocks. This allowed the circuitry

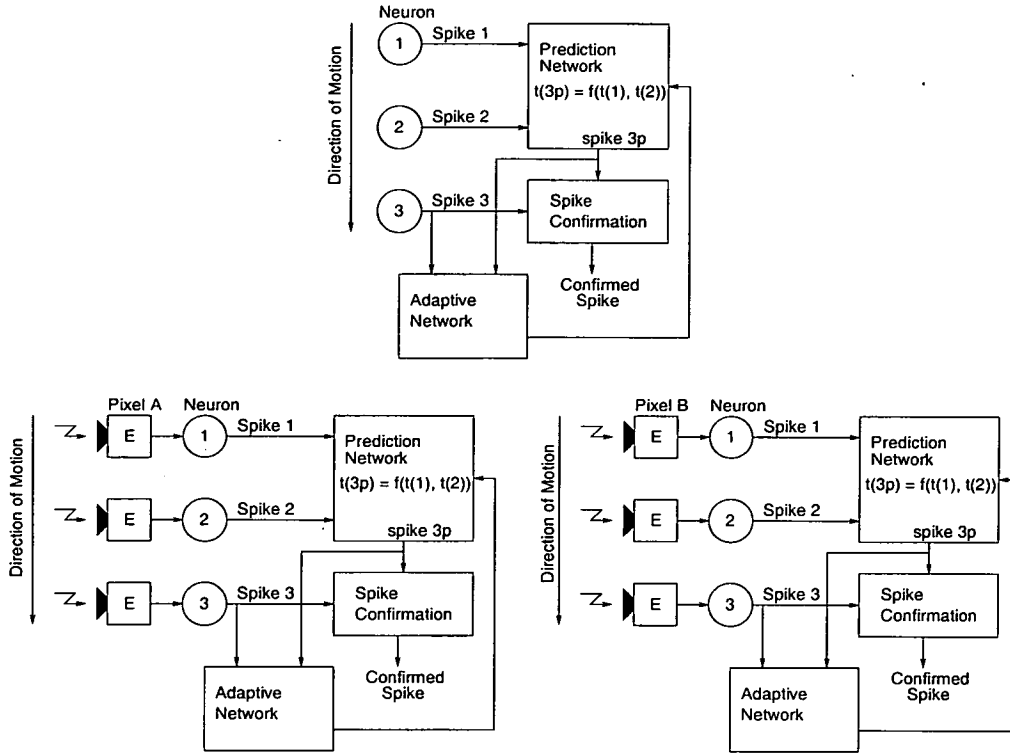


Figure 5.6: Architecture of the first chip fabricated. Three prediction/adaptation blocks were laid out. One had voltage controlled inputs, the other two were pixel activated.

to be tested independently of the pixels.

Section 5.5 details results from the voltage controlled prediction/adaptation block. The voltage spike inputs were generated by the Hewlett Packard 16522A pattern generator. The prediction error was measured using the crosshair markers on the Hewlett Packard oscilloscope 16534A. Section 5.6 reports results from the pixel activated circuitry. These results were recorded by Vasin Boonsobhak using an Agilent 54624A oscilloscope. Both sets of results were recorded to a $1\mu\text{s}$ accuracy.

5.5 Chip Results

The circuitry was fabricated, using the AMS $0.35\mu\text{m}$ CSI process. The initial tests were performed on the circuitry with voltage controlled inputs. This was also the circuitry that was tested with varying V_{dd_mirr} , see figure 4.5.

The oscilloscope traces in figure 5.7 are taken from one of the chips when the input neurons

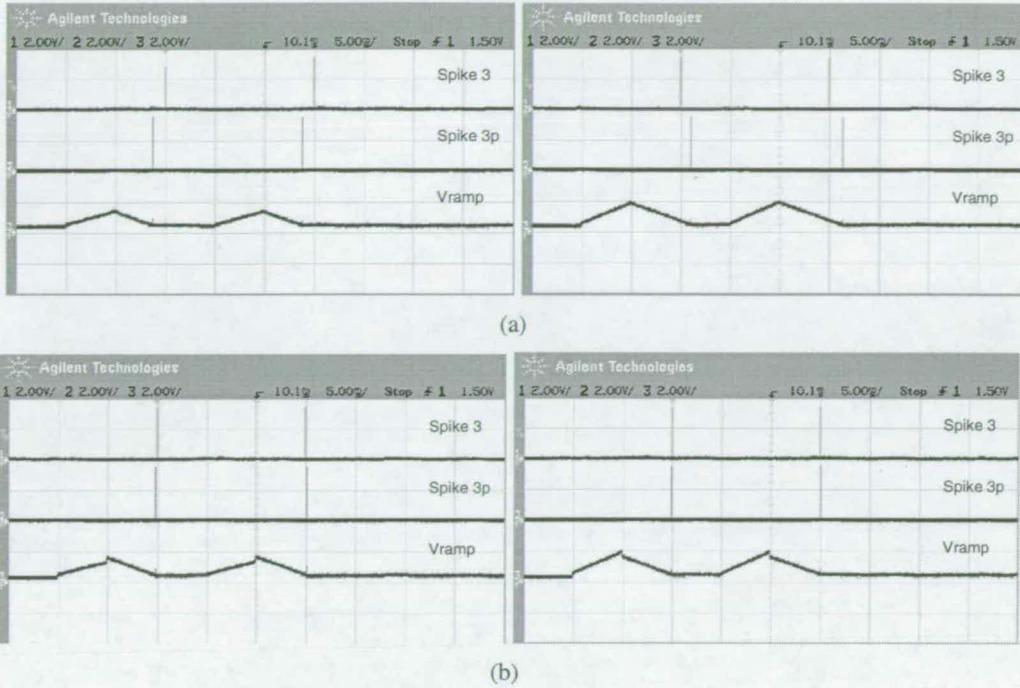


Figure 5.7: Pre- and post-adaptation results. These oscilloscope traces show the spike 3, spike 3p and Vram signals during testing. At $t(1)$ Vram is set to Vrampth. The timescale is 5ms/div. (a) Pre-adaptation and the error in spike 3p is clear. (b) The signals post-adaptation where the error has been reduced. The alteration to Vram can also be seen.

were fired with a 5ms inter-spike interval, $t(2) - t(1)$. It can be seen that the spike timing dependent adaptation improves the prediction significantly. Figure 5.8 shows part of the adaptive process. As the weight increases the resultant improvement in the prediction can be seen.

The circuit was further tested by activating the three input neurons with 1ms and 5ms inter-spike intervals and measuring the prediction error, $t(3p) - t(3)$ before and after adaptation. This was done three times on eight different chips with V_{dd_mirr} varying between 3.28 and 3.31V. Figure 5.9 shows the pre and post adaptation error when the inter-spike interval is 1ms, (a), and 5ms, (b). These results compare favourably with the simulation results shown in figure 5.5.

5.6 A Small Adaptive Pixel Array

After operation was confirmed using the voltage controlled inputs, testing began on the pixel activated circuitry. The generation of spikes from the pixel input had been verified and the

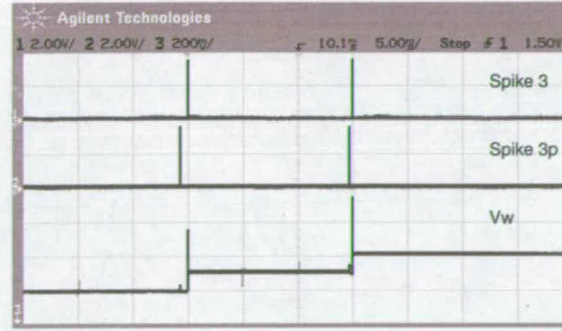


Figure 5.8: Oscilloscope trace showing weight change. The difference in spike timing alters the synaptic weight and the prediction becomes more accurate. The timescale shown is 5ms/div

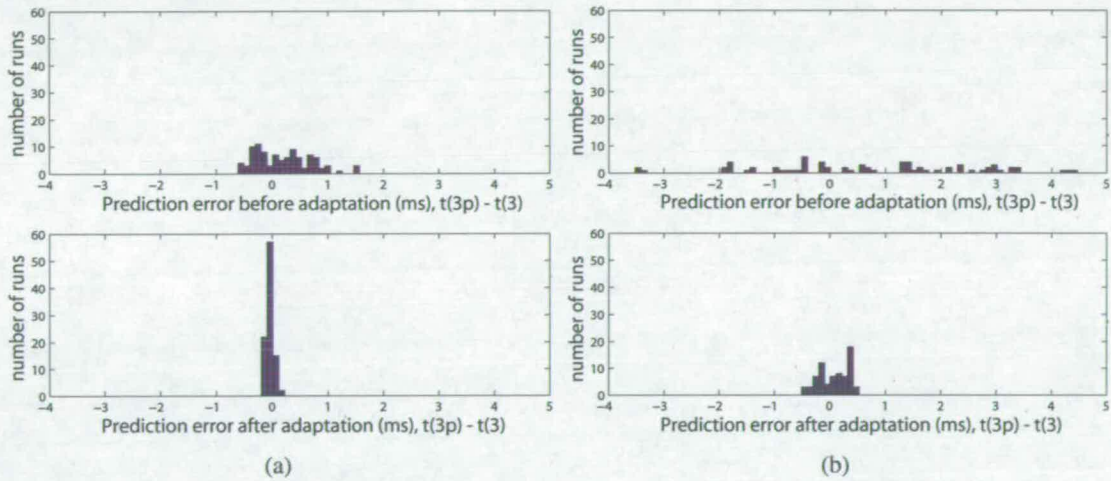


Figure 5.9: Distribution of prediction errors. The prediction error is defined as $t(3p) - t(3)$. Each graph has the pre-adaptation results above the post-adaptation ones. The results were obtained with V_{dd_mirr} varying between 3.28 and 3.31V Results (a) were when $t(2) - t(1) = 1ms$ and (b) $t(2) - t(1) = 5ms$

prediction delay was measured before and after adaptation. For this test V_{dd_mirr} was kept at 3.3V and the inter-spike intervals used were 1ms and 5ms. The results were taken from four chips, twenty results from each, rather than the eight for the voltage driven circuitry. Figure 5.10 shows the resultant spread of the errors. It can be seen that adaptation improves the prediction substantially. When these results are compared with those taken from the voltage controlled circuit when V_{dd_mirr} was 3.3V, figure 5.11, it can be seen that the pixel does not affect the spread of error. The mean prediction error and the standard deviation was measured for each distribution and can be found in table 5.1.

The similarity of the numbers in table 5.1 between the pixel and voltage driven circuitry con-

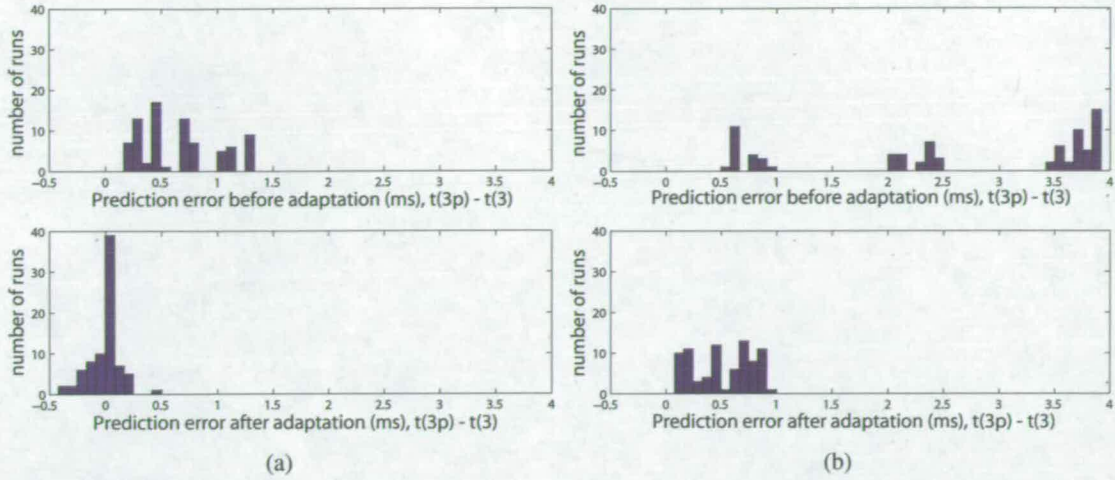


Figure 5.10: Distribution of prediction errors from pixel activated circuit. The error measurements were taken with $V_{dd_mirr} = 3.3V$. The current integrated on the leaky integrate-and-fire neurons came from transient-detecting pixels. Results (a) were from tests when $t(2) - t(1) = 1ms$ and (b) $t(2) - t(1) = 5ms$

firmly that pixel activation does not introduce additional error. Small differences should be expected as the results were collected from a smaller sample of die.

The error when the inter-spike interval is 5ms will always be greater than the 1ms error, because the prediction error is related to the mismatch in current and charging time. The pre-adaptation mean values shown in table 5.1 can be seen to have a positive bias. Transistors N1 and N3 in figure 4.5 will be affected by the short channel effect, and this would cause a small positive bias, but simulation results indicate it should not be as big as the value seen. It is suspected that the majority of the bias was caused by connecting the drains of the current mirror, V_{dd} and V_{dd_mirr} , externally to the chip, rather than with a direct metal connection. A voltage drop of a few mVs on the V_{dd} line would explain this systematic offset. This chip is therefore not suitable for determining the level of mismatch in this process, but it does illustrate the effectiveness of a spike timing dependent adaptive approach for minimising the effect of mismatch. The adaptation does move the mean back towards zero, and the standard deviation is greatly reduced.

Two different types of error are included in the prediction: an offset error and a gain error. The offset error is introduced by the non-infinite gain of the comparator in figure 4.5 and the short channel effect, whereas the gain error is due to mismatch in the $\frac{I_D}{V_{gs}}$ of the current mirror transistors at the dc bias point set by V_b . Monte Carlo simulations indicated that differences in

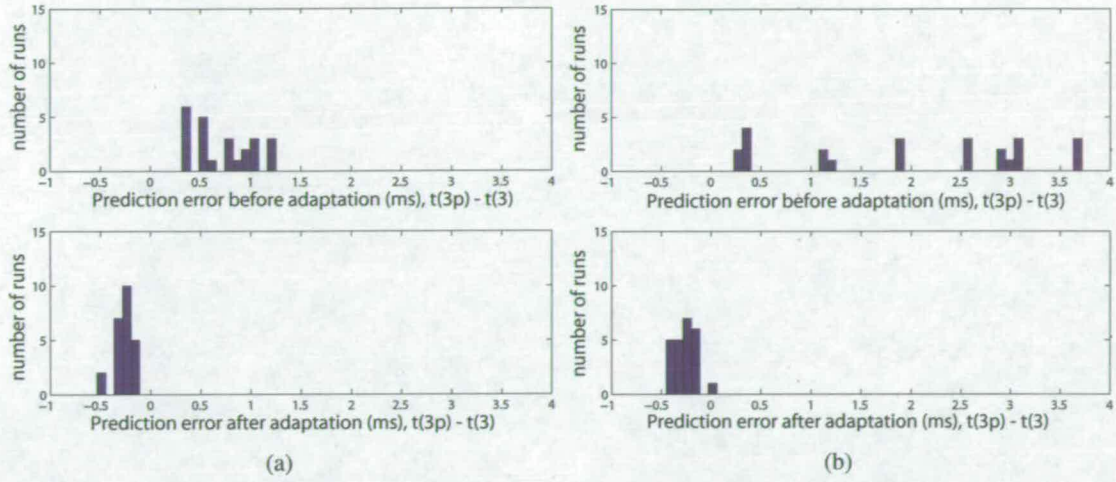


Figure 5.11: *Distribution of prediction errors from the voltage controlled circuit. These results were recorded when $V_{dd_mirr} = 3.3V$. Results (a) were from tests when $t(2) - t(1) = 1ms$ and (b) $t(2) - t(1) = 5ms$*

gain would be the dominant source of prediction error. This can be observed as the difference between the rate of charging and discharging of V_{ramp} , figure 4.6.

This particular approach for prediction correction will correct for both types of error but only for one time interval at a time, as it essentially applies an offset correction. Therefore if a new time delay is introduced, the adaptive circuit must move to a new weight value. The next chapter describes a way to implement the correction by adjusting the current sources in the current sink/source circuit (figure 4.5), so that the prediction can be correct over a range of time intervals. It is also worth noting that should a systematic error occur during spike generation at the pixels, the prediction will adapt to accept that error. A random error will have no effect.

5.7 Summary and Conclusions

This chapter has described circuitry used to improve a prediction of a spike event based on previous events. An excitatory/inhibitory synapse previously designed in this university was used to inject or remove charge from an amount stored across a capacitor which represented the prediction. This allowed the prediction to be brought forward or delayed in time.

Circuitry was needed to alter the synaptic weight voltage and therefore the amount of charge injection/removal. A spike timing dependent algorithm was used to adapt the weight to the point where the prediction and actual spike became approximately coincident.

	Before Adaptation		After Adaptation	
	Mean (ms)	Std (ms)	Mean (ms)	Std (ms)
Chip Simulation 1ms delay	-0.14	0.21	0.13	0.06
Chip Simulation 5ms delay	-0.21	0.50	0.11	0.09
Pixel Driven 1ms Delay	0.65	0.36	-0.01	0.13
Pixel Driven 5ms Delay	2.62	1.27	0.52	0.27
Voltage Driven 1ms Delay	0.72	0.31	-0.27	0.08
Voltage Driven 5ms Delay	1.99	1.23	-0.26	0.11
Varying <i>Vdd_mirr</i> 1ms delay	0.20	0.50	-0.04	0.06
Varying <i>Vdd_mirr</i> 5ms delay	0.71	1.90	-0.084	0.27

Table 5.1: Mean and standard deviation of prediction error for design one.

The circuitry was fabricated using the AMS $0.35\mu\text{m}$ CSI process and tested by applying spike patterns with 1 and 5ms between spikes. The circuitry was allowed to adapt and the prediction error was measured. This could then be compared to the accuracy of the prediction when the adaptation was turned off.

The results showed that the spike timing dependent adaptation could indeed improve the prediction, with the standard deviation of the prediction errors improving from 1.23 to 0.11ms, for a 5ms time delay between spikes when stimulated by voltage spikes from a pattern generator. When a 1ms time delay was applied the improvement was still significant but smaller, as the starting prediction was more accurate. The circuitry was also successfully activated by pixel input without degrading the performance.

While the adaptive circuitry could improve the prediction performance it did have a number of significant flaws, the most important of which is that the synaptic weight learned, and therefore the amount of prediction alteration applied, is only correct for one particular time delay. The prediction element is part of a larger depth-from-motion algorithm and a key point is that different depths can be identified by their different time delays. Therefore while the adaptive circuitry has shown that spike timing dependent adaptation can be used to compensate for mismatch, a more sophisticated system of adaptation is required to achieve the desired outcome of actually matching the currents. It would also be desirable to improve the resolution at which the prediction adaptation can operate, resulting in a narrow post-adaptation range of results. These concerns are addressed in the next chapter.

Chapter 6

An Improved Adaptive Technique

6.1 Introduction

The original design for the adaptive circuitry, described in the previous chapter, suffered from the problem that most of the mismatch was caused by a gain error, i.e. differing $\frac{I_D}{V_{gs}}$ at a fixed dc bias point, but no gain correction could be applied. Instead an offset correction, equivalent to addition or subtraction of time, was applied. This correction was only applicable when the inter-spike interval was constant, in that case 5ms. When the interval was changed to 1ms, the circuit had to adapt to a different level of correction.

While the design was sufficient to show that spike timing could be used to minimise the effect of mismatch and improve the prediction, it was within very restricted operating conditions. For the second design, the performance as the inter-spike interval changed had to be as good as that achievable with a dynamic current mirror, section 4.3. It also had to have the ability to predict a $t(2) - t(1) : t(3) - t(2)$ ratio other than 1:1.

6.2 The Redesigned Prediction Circuitry

For the first circuit design the choice was made to compensate for the effect in the difference in the currents, rather than adjusting one of the currents, section 5.1. The predictive circuitry was re-designed to address this by providing the ability to adjust the discharging current, and is shown in figure 6.1. The sequence of operation is:

- t(1) Spike 1 is fired and N6 acts as a switch which sets the initial value of V_{ramp} to V_{rampth} .
- t(1)-t(2) P5 is on and C1 is charged through P3 and P4 which supplies a mirrored version of the current set by V_b .
- t(2) P5 turns off while N5 becomes active.



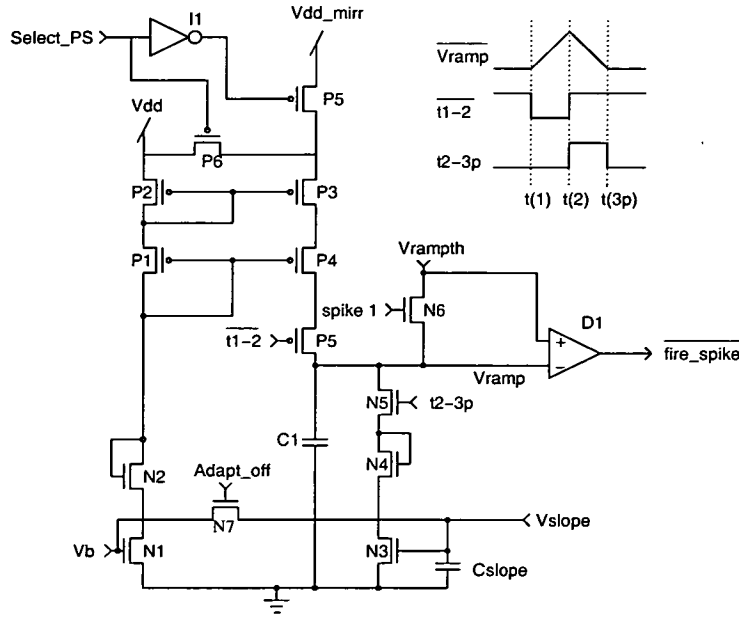


Figure 6.1: Spike prediction circuit. $C1$ charges between $t(1)$ and $t(2)$ and discharges from $t(2)$ to $t(3p)$. If N7 is closed V_b equals V_{slope} and an accurate prediction will be given if the transistors have good matching. If that is not the case N7 can be opened and V_{slope} can be set at a different voltage to ensure equal charge and discharge times. The Select_PS signal allows for one side of the current mirror to be connected to a signal other than Vdd. This enables mismatch to be forced on to the chip for testing purposes

- $t(2)-t(3p)$ $C1$ is discharged through N3 and N4 at a rate set by the voltage across C_{slope} . If the current source and sink are matched and V_b equals V_{slope} the time taken to discharge $C1$ should match the charging time.
- $t(3p)$ V_{ramp} is compared to V_{rampth} using a differential pair. When V_{ramp} crosses it, neuron 3p is “fired”. This ends the discharge period.

The bias voltage applied to N1 is now no longer permanently connected to N3. The closing of N7 allows the voltage across C_{slope} to be initialised to V_b , but it can be open to enable different bias voltages to be set. The W/L ratio for transistors N3 and N4, and therefore N1 and N2, was altered to reflect the driving of N3 by the voltage across C_{slope} . The length of the transistors determines their response to the changing voltage V_{ramp} and was set at $3\mu m$. To reduce the capacitive coupling of V_{ramp} to V_{slope} it was important to keep the gate-drain parasitic capacitance to a minimum. The width chosen to reduce this effect was $0.6\mu m$.

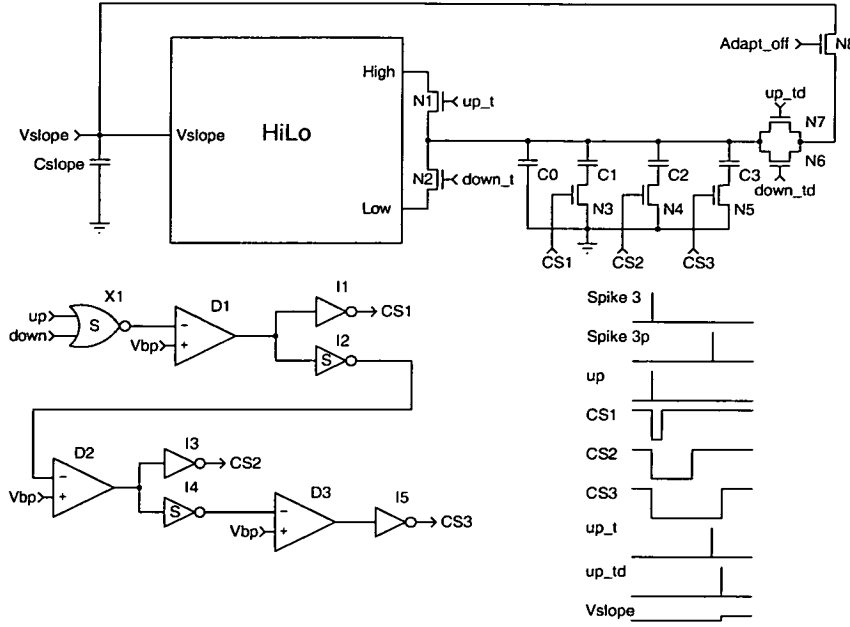


Figure 6.2: Vslope update circuitry. The circuit employs a switched capacitor technique to increase or decrease the voltage V_{slope} . The amount of change is determined by the signals CS1-3 which turn on over time, one after another. The logic gates marked S have a slow rise time and implement the delay between CS signals. The configuration of the HiLo circuitry is shown in figure 6.3.

If N7 is closed, the prediction accuracy demands that the matching between transistors be good. Should mismatch occur, N7 can be opened and V_{slope} can be altered until the prediction becomes accurate. The alteration is performed by the adaptive circuitry described in the next section, based on the time difference between spikes 3 and 3p. The circuitry used to generate control signals $\overline{t1 - 2}$ and $t2 - 3p$ is shown in appendix C, figure C.1. It is more complicated than the circuitry used for the previous chip, in that it contains additional logic to ensure noisy spikes do not cause, for example, $\overline{t1 - 2}$ and $t2 - 3p$ to be active at the same time.

As only 20 chips will be received and the amount of mismatch is not guaranteed, the prediction circuitry was designed so that mismatch could be forced on to the current mirror. If $Select_PS$ is low both sides of the current mirror are connected to V_{dd} . When it is high, one side is connected to V_{dd_mirr} . This voltage can then be altered to change the current levels through the charging side of the circuitry. This design provides a local connection between V_{dd} and V_{dd_mirr} so that when the two voltages are supposed to be equal they are.

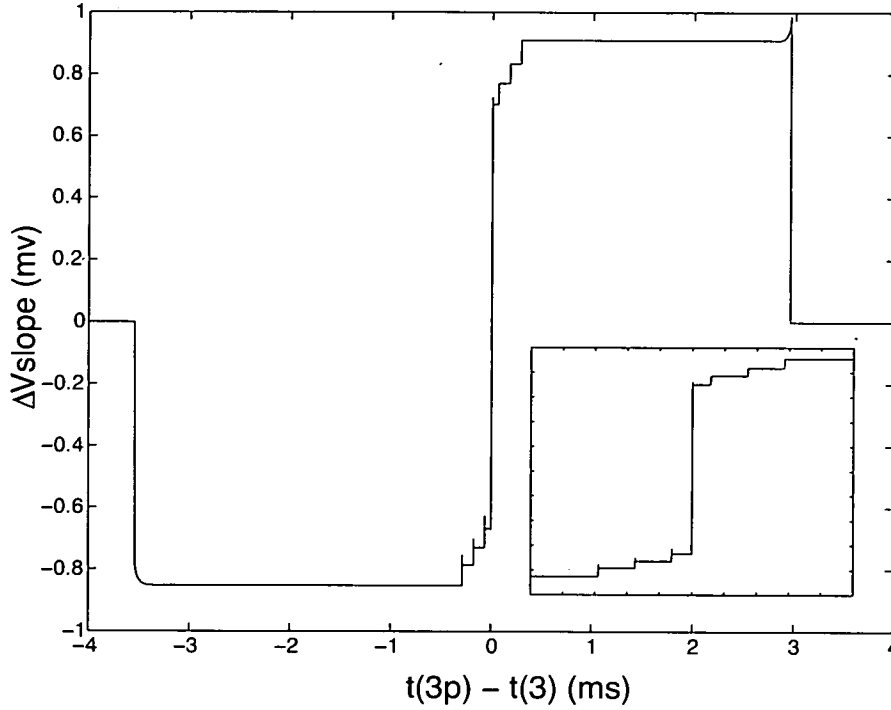


Figure 6.4: *The change in Vslope with prediction error. The change in voltage of Vslope implemented by the circuitry shown in figures 6.2 and 6.3. The inset graph is of the range $-500\mu s - 500\mu s$ and shows the effect of switching in the capacitors C1-3 in figure 6.2.*

amount, either up or down, irrespective of the accuracy of the prediction. This would lead to a choice between either very slow convergence and high accuracy or fast convergence with low accuracy. The inclusion of the capacitors is controlled by $CS1-3$ which are reset when the up signal is generated at the arrival of spike 3. The NOR and NOT gates marked S are “current starved”, by a subthreshold biased transistor, connected between V_{dd} and the standard circuit configuration, (see appendix B,) and therefore have a slow rise time. After up returns to signal ground, the output of the NOR gate slowly rises until it reaches V_{bp} , in this case 2.5V. At this point, $CS1$ becomes high and the output of the first slow NOT gate starts to rise. With $CS1$ high charge can be stored on the connected capacitor. $CS2$ and $CS3$ become high approximately $100\mu s$ after the preceding CS signal. When spike $3p$ arrives up_{\downarrow} is generated and the *High* voltage from the *HiLo* circuit is stored across all active capacitors. Therefore the amount of stored charge is related to the time between spike 3 and spike $3p$. Up_{\downarrow} is generated slightly later at which point the charge is shared between the capacitor bank and C_{slope} , resulting in a rise in V_{slope} . C_{slope} is a much larger capacitor than the others. Therefore the voltage change is much smaller than the 200mV difference generated by the *HiLo* circuitry. Figure 6.4 shows the

resulting change of V_{slope} with different inter-spike intervals between the actual and predicted spikes. A windowing mechanism is used so that if a predicted spike arrives much earlier or later than the actual spike, they are deemed to be unrelated. This results in the zero change at $|t(3p) - t(3)| > 3ms$. This is less than for the circuitry described in the previous chapter, as V_{slope} can be initialised to V_b where as V_w , figure 5.3, initially starts at zero which produces a close to 100% prediction error. As the adaptive network drives V_{slope} towards a particular point it can be left active after calibration. Therefore this adaptation mechanism retains the ability to compensate for any circuit drift caused by changes in environmental conditions or charge leakage from the capacitor. The only constraint is that more correct spikes are present than incorrect ones.

6.4 Chip Architecture

In the first chip three copies of the same circuitry were laid out two of which connected to pixels designed by another student. As it had been shown that pixels could fire the neurons without affecting performance, the second adaptive chip did not include any pixels. Instead five neurons were laid out, and three prediction and adaptive circuits were connected to them as shown in figure 6.5. This is closer to the way a full system would have to be set up where an edge is traced along a retinal radius. The prediction of spike 5 has been set so that $t(5p) = t(4) + \frac{6}{5}\{t(4) - t(3)\}$. This is done by setting the width of transistors N3 and 4 to $0.5\mu m$ compared to the $0.6\mu m$ for transistors N1 and 2 in figure 6.1.

To allow a comparison with one of the other prediction methods shown in chapter 4, a dynamic current mirror prediction circuit was designed and laid out. The circuit design is the same as that shown in figure 4.3 with the exception that P4 was removed. The dynamic current mirror prediction cannot do $t(5p) = t(4) + \frac{6}{5}\{t(4) - t(3)\}$.

6.5 Results

The circuitry was designed using the AMS $0.35\mu m$ C35 process. The CSI process used for the first chip was no longer supported. The spike inputs were generated by the Hewlett Packard pattern generator 16522A, and the output spikes were recorded by the Hewlett Packard logic analyser 16555D, sampling at 2.5MHz. The accuracy of the recorded spike time was restricted

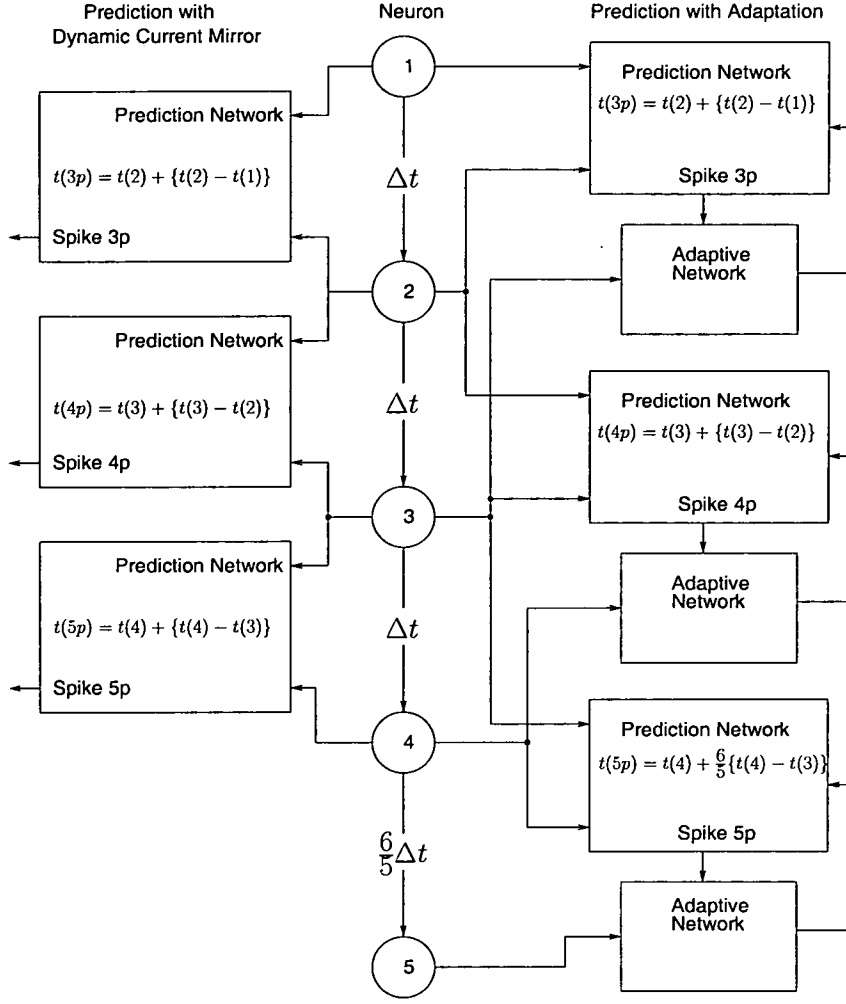


Figure 6.5: Architecture of the second chip fabricated. Five neurons were laid out and used to drive six prediction networks, three with spike timing dependent adaptation and three using dynamic current mirror circuitry. The interval between the spikes was set at: $t(2) - t(1) = t(3) - t(2) = t(4) - t(3) = \frac{5}{6} \{t(5) - t(4)\}$.

to the time stamp accuracy which was $1\mu s$.

6.5.1 Static Time Delays

Initially a spike pattern where $t(2) - t(1) = t(3) - t(2) = t(4) - t(3) = 5ms$ and $t(5) - t(4) = 6ms$ was applied 33 times with a new cycle beginning every 25ms. The output spike times were recorded and the prediction errors, $t(3p) - t(3)$, $t(4p) - t(4)$, and $t(5p) - t(5)$, were calculated. Example traces, recorded from 5 different chips, are shown in figure 6.6. The results for each adaptive block are shown independently for both the prediction with spike timing dependent

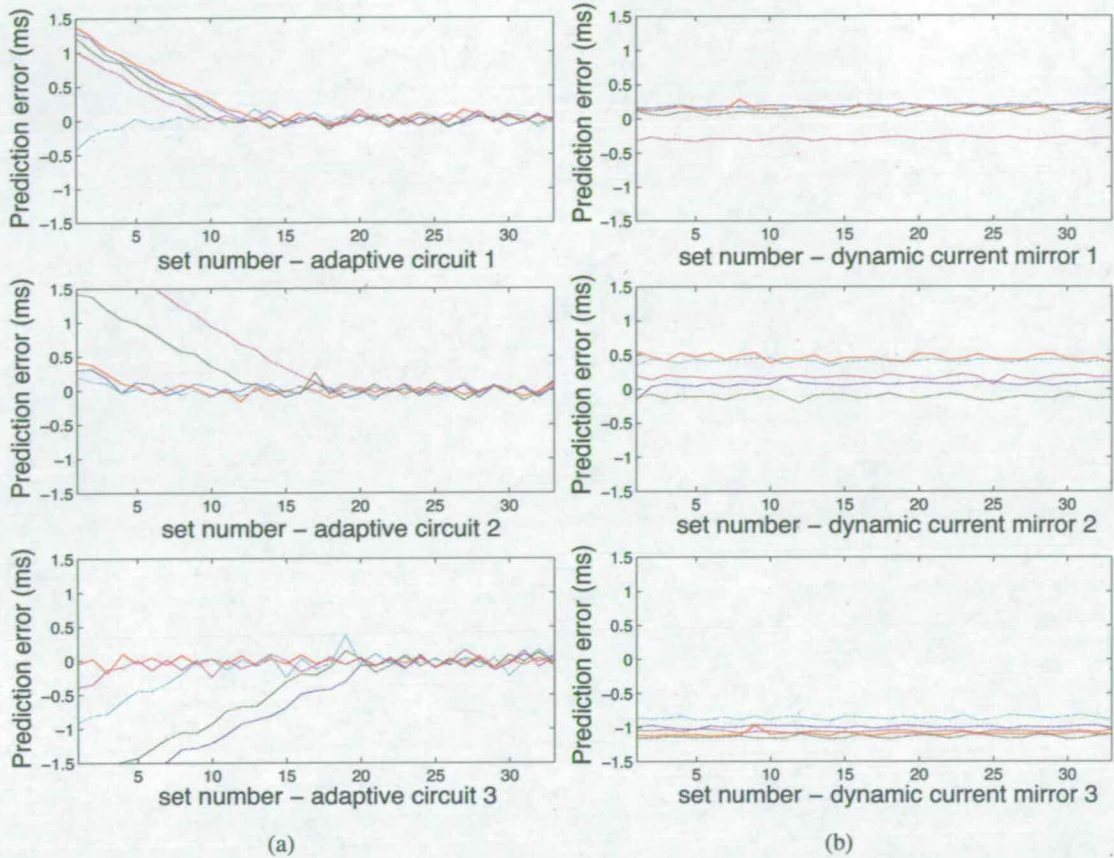


Figure 6.6: Prediction error during adaptation. 33 spike sets are applied to the circuitry and the prediction error measured. Each graph shows the result from 5 chips, a different colour for each. (a) Shows the prediction error, $t(3p) - t(3)$, $t(4p) - t(4)$, and $t(5p) - t(5)$, for the circuitry with adaptation. (b) shows the prediction error for the dynamic current mirror circuitry.

adaptation, and the dynamic current mirror. *Adaptive circuit 1* corresponds to the prediction with adaptation of spike 3 (figure 6.5), *dynamic current mirror 2* refers to the prediction of spike 4 using the dynamic current mirror circuitry.

Figure 6.6(a) shows the error as the circuitry adapts, and it can be seen that all three prediction/adaptation blocks perform well. When the error approaches zero, the error begins to oscillate around zero. This is a direct result of the shape of the voltage change curve in figure 6.4. Around $t(3p) - t(3) = 0$ there is a swing of approximately 1.4mV, measured in simulation. This means that when the prediction is close to $t(3)$ it tends to oscillate between “too fast” and “too slow” but is unable to get exactly to the zero point. It can be seen that when the prediction ratio is 1:1.2, the circuitry performs just as well as the 1:1 predictions.

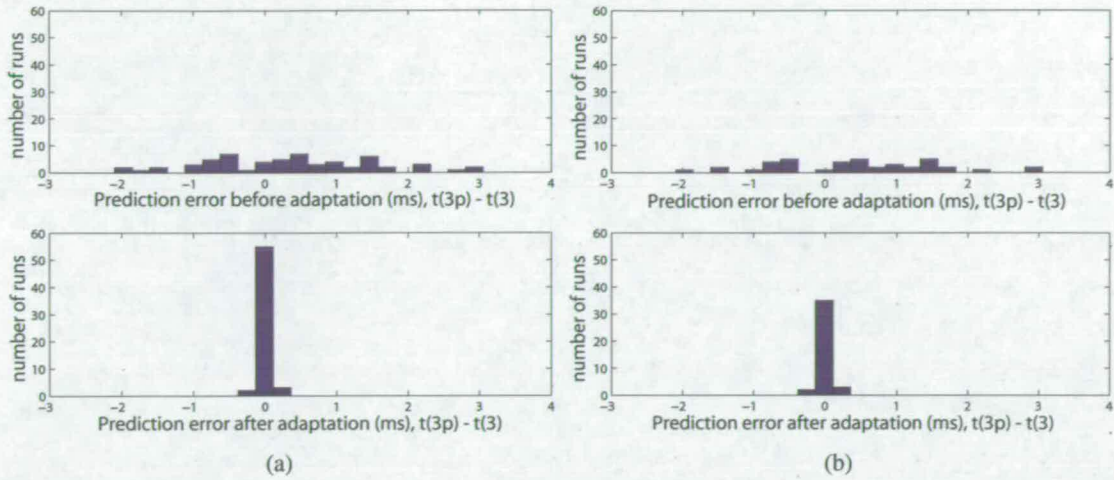


Figure 6.7: Distribution of prediction errors - with adaptation. The time differences, $t(3p) - t(3)$, $t(4p) - t(4)$, and $t(5p) - t(5)$ were measured before and after adaptation. In (a) the results for all three prediction/adaptation blocks are combined together. In (b) only the results where the prediction ratio is 1:1 are used.

As expected the dynamic current mirror performance, figure 6.6(b), stays approximately constant over the 33 spike sets. The error $t(5p) - t(5)$ is particularly large as the dynamic current mirror could not predict the 1:1.2 ratio. Just from this small set of results, it can be seen that the adaptive circuitry can perform better than the dynamic current mirror.

The test was performed on the 20 chips received and a comparison of the error before and after adaptation is shown in 6.7(a). The 3 results from each chip are combined into the single histogram. The positive bias visible in figure 5.11 is no longer visible. This justifies the inclusion of transistors P5 and P6 in figure 6.1, which ensures that when the supply voltages for both sides of the current mirror are to be at the same potential, there is a local connection rather than just a connection outside the chip. The results are far superior to those from the earlier chip. Figure 6.7(b) shows only the results from the 1:1 predictions. Comparing (a) and (b) it can be seen that the 1:1.2 prediction does not produce significantly different results.

Figure 6.8 compares the results post-adaptation to those from the dynamic current mirror. In (a) there are two peaks in the distribution of the dynamic current mirror results. The largest, with a mean of approximately $100\mu s$ results from the predictions of spikes 3 and 4. The prediction of spike 5 causes the smaller distribution with a mean of $-840\mu s$. The adaptive circuitry performs better even if the comparison is only made between the 1:1 ratio predictions, figure 6.8(b). In [122] various methods to improve the performance of dynamic current mirrors are described.

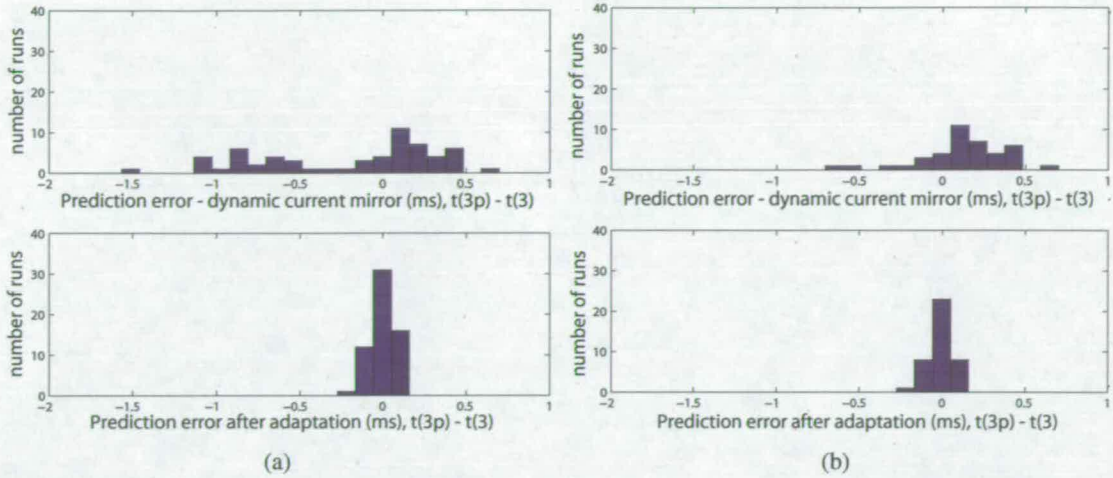


Figure 6.8: Distribution of prediction errors - dynamic current mirror. The time differences, $t(3p) - t(3)$, $t(4p) - t(4)$, and $t(5p) - t(5)$ were measured for the dynamic current mirror circuitry and are compared to the post-adaptation results. In (a) the results for all three prediction/adaptation blocks are combined together. In (b) only the results where the prediction ratio is 1:1 are used.

These include very basic ones such as increasing the size of the capacitor C2, (see figure 4.3), to more complicated methods of cancelling out the effects of switching noise. While the size of C1 was quite large, 1pF, no other steps were taken to improve the performance of the dynamic current mirror. Applying them may improve the performance of the dynamic current mirror to the level of the post-adaptation circuitry, or even better, but the difficulty of making 1:x predictions would still remain.

After performance had been verified with V_{dd} connected to both sides of the current mirror, *Select_PS* was changed so that V_{dd_mirr} was connected. V_{dd_mirr} was set at 3.25, 3.255 and 3.26V and the prediction error before and after adaptation was measured. If V_{dd_mirr} was set at 3.3V, the nominal power supply, V_{ramp} charged at such a high rate that it saturated at the power supply within the 5ms. To regain operation similar to that when tied to V_{dd} , V_{dd_mirr} had to be dropped to approximately 3.25V. This reinforces the idea that a drop across the power supply lines caused the positive bias in figure 5.11.

The results are shown in figure 6.9. It can be seen that a few prediction errors are not improved by adaptation. This is because the original error was outside the window of adaptation. The performance within the window of adaptation remains good.

A summary of the mean and standard deviation for the circuitry can be found in table 6.1. The

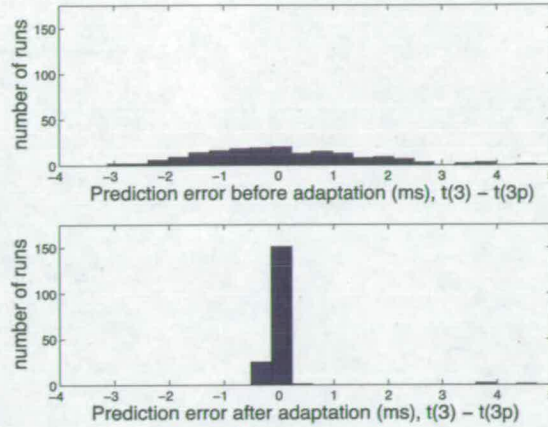


Figure 6.9: *Distribution of prediction errors with varying V_{dd_mirr} . The time differences, $t(3p) - t(3)$, $t(4p) - t(4)$, and $t(5p) - t(5)$ were measured before and after adaptation while V_{dd_mirr} was set at 3.25, 3.255 and 3.26V. A few predictions did not improve as the original error was outside the adaptation window.*

results post-adaptation for this chip are significantly better than those from the dynamic current mirror and the first chip, see table 5.1. The large post-adaptation standard deviation for the varying V_{dd_mirr} is caused mostly by the three runs where the error was outside the adaptation range.

6.5.2 Varying Inter-Spike Intervals

Having assessed the ability of the circuitry to improve performance, it was important to establish whether the change in adaptation method allowed for improved performance when varying inter-spike intervals were applied. Within the context of the larger algorithm different intervals correspond to edges at different depths. Therefore it is important to be able to respond to this variation.

Two spike trains were set up with varying intervals. The first was 32 spike sets long, and consisted of a smaller 16 spike sets grouping repeated twice. A spike set is a grouping of spikes 1, 2, 3, 4 and 5. The interval was varied between 1 and 6ms with the delay between spikes 4 and 5 being 1.2 times that. The second spike train consisted of 33 spike sets again varying between 1 and 6ms. Table 6.2 lists the times for Δt .

Figure 6.10 shows the results of these simulations for each of the three blocks fabricated. The 32 spike set simulation is shown in (a). Where the large changes in inter-spike interval occur, spikes in the error can be seen, although the second time the circuitry sees the pattern, the effect

	Before Adaptation		After Adaptation	
	Mean (ms)	Std (ms)	Mean (ms)	Std (ms)
1:1 prediction with adaptation simulation	0.29	0.58	-0.01	0.05
1:1.2 prediction with adaptation simulation	-0.24	0.55	-0.08	0.07
1st predictive block with adaptation	0.16	1.13	0.01	0.08
2nd predictive block with adaptation	0.71	1.12	-0.03	0.06
3rd predictive block with adaptation	0.18	1.26	0.01	0.01
Combined result for prediction with adaptation	0.35	1.18	0.00	0.07
Prediction with adaptation and varying V_{dd_mirr}	0.08	1.45	0.05	0.53
Dynamic current mirror prediction 1	0.10	0.26		
Dynamic current mirror prediction 2	0.16	0.28		
Dynamic current mirror prediction 3	-0.84	0.26		

Table 6.1: Mean and standard deviation of prediction error for design two.

is reduced. The results in (b) are more stable as the Δt for the early spike sets stays close to 5ms. The only serious deviation from the desired result occurs when the delay jumps from 1 to 6ms. There is no significant difference between the circuitry designed to have a prediction ratio of 1:1 and 1:1.2 for either spike train.

The results from the two 1:1 prediction blocks were then combined so that they could be compared to results from the dynamic current mirror prediction circuitry and simulations of the original adaptive circuitry. The poor initial matching of the original adaptive circuitry meant it would be misleading to compare it directly with the new circuit. Therefore the comparison was done against simulation results. The simulation performed better than the fabricated circuitry. Therefore any improvement in performance seen between the two adaptive circuits is actually greater than that shown. As the dynamic current mirror circuitry could not predict the 1:1.2 predictions, these are also removed from the comparison. The results are shown in figure 6.11, (a) is for the 32 spike sets and (b) is for the 33. After the initial period of adaptation, it can be seen that for both spike trains, the improved adaptive circuitry performs better than both the original adaptive circuitry and the dynamic current mirror. The performance of the dynamic current mirror prediction can be seen to degrade as the prediction time becomes larger. This is particularly visible in (a) where the time delay increases and then falls. This behaviour can be seen in the adaptive version too, but by the second repetition the change has been significantly reduced.

While the gain correction provides a better result over changing time delays than the offset

Spike Set 1		Spike Set 2	
Δt (ms)	Repetitions	Δt (ms)	Repetitions
1	1	5	6
2	1	4	6
3	1	6	6
4	1	5	6
5	1	3	1
6	1	4	1
5	1	2	1
4	1	1	1
3	1	6	1
2	1	4	1
1	1	5	1
6	1	3	1
2	1	5	1
5	1		
3	1		
4	1		

Table 6.2: Delay times applied for variable delay tests. Spike Set 1 was applied twice.

correction, it is still not perfect. This is so because the error in the prediction is composed of both a gain and an offset error.

6.5.3 The Effect of Noisy Spikes on Adaptation

The prediction element of the depth-from-motion algorithm was added to reduce the effect of noise on the depth calculation. It is therefore important that the adaptive circuitry continues to work even if there is noise. In section 5.2 it was assumed that if accurate predictions outnumbered incorrect predictions the adaptation would not be effected in the long-term. With the second chip this assumption was tested.

The noise on the input spike train can come from, for example, camera shake. Unfortunately no data was available as to the likely frequency or position noise would appear in the spike train. Therefore, a random generation of noise was performed with one noisy spike being generated for each genuine spike. The results shown in figure 6.7 were recorded when the inter-spike interval was 5ms and the cycle of firings was repeated every 25ms. As the pattern generator applied a new pattern every 200 μ s this meant the cycle was repeated every 125 patterns. To add noise a spike was generated using the MATLAB *rand* function. This function generates a

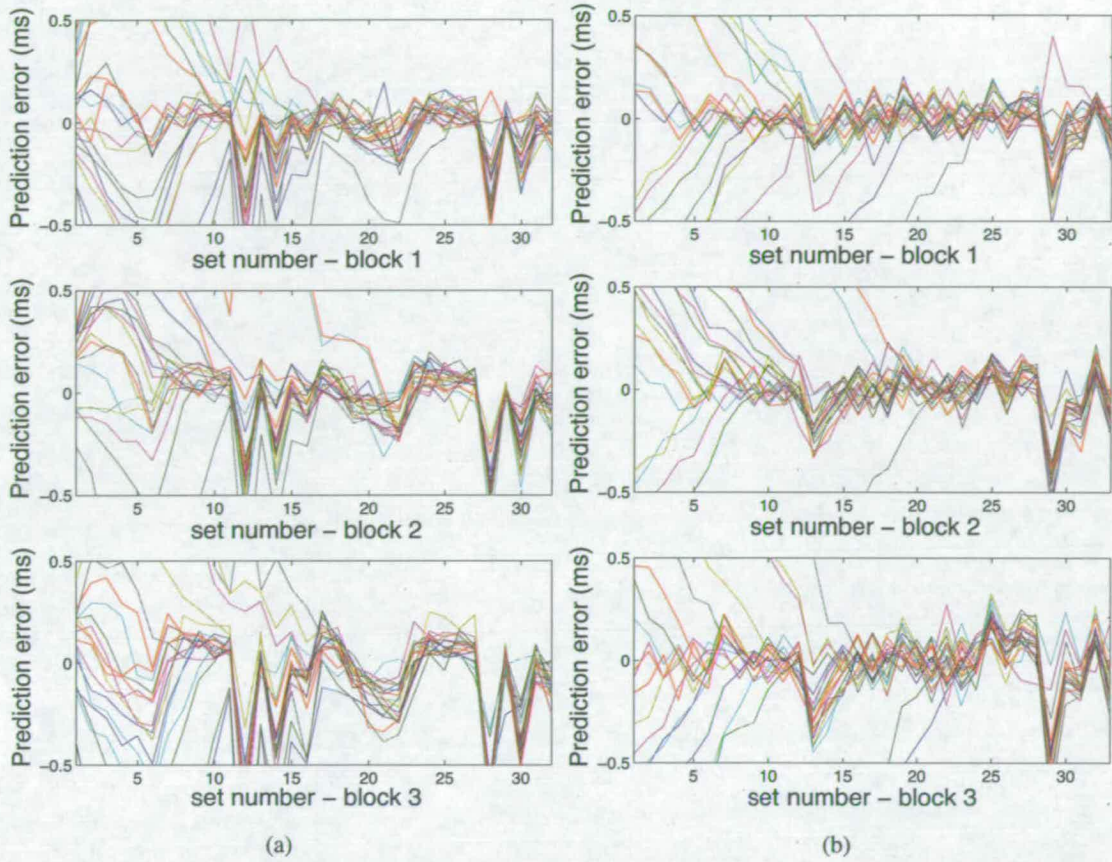


Figure 6.10: *Prediction error during adaptation with varying inter-spike interval for three blocks of adaptive circuitry. Spike trains with varying inter-spike intervals were applied to neurons 1 to 5 and the prediction error for blocks 1, 2 and 3 was measured. (a) a spike train with 16 spike sets repeated twice. (b) a spike train with 33 spike sets.*

pseudorandom number between zero and one. It was then multiplied by 125 and rounded to the nearest whole number. A spike was then placed in this position for neuron 1 during cycle 1. This was repeated so that each of the 33 cycles for all 5 neurons contained a noisy spike as well as the genuine spike.

The noisy spikes can have various effects. The control circuitry for the prediction network (Appendix C, figure C.1) was designed to ensure that $\overline{t1 - 2}$ and $t2 - 3p$ in figure 6.1 could not be active at the same time. $t(3p) = t(2) + \{t(2) - t(1)\}$, for a 1:1 calculation, therefore the only signals that can affect the calculation of a prediction are spikes 1 and 2, and in addition the noisy spikes must occur in the interval $t1 - 2$. If the noise is on spike 1, the prediction will be incorrect by $t(1)_{noise} - t(1)_{actual}$. The effect of noise on spike 2 is greater at $2(t(2)_{actual} - t(2)_{noise})$.

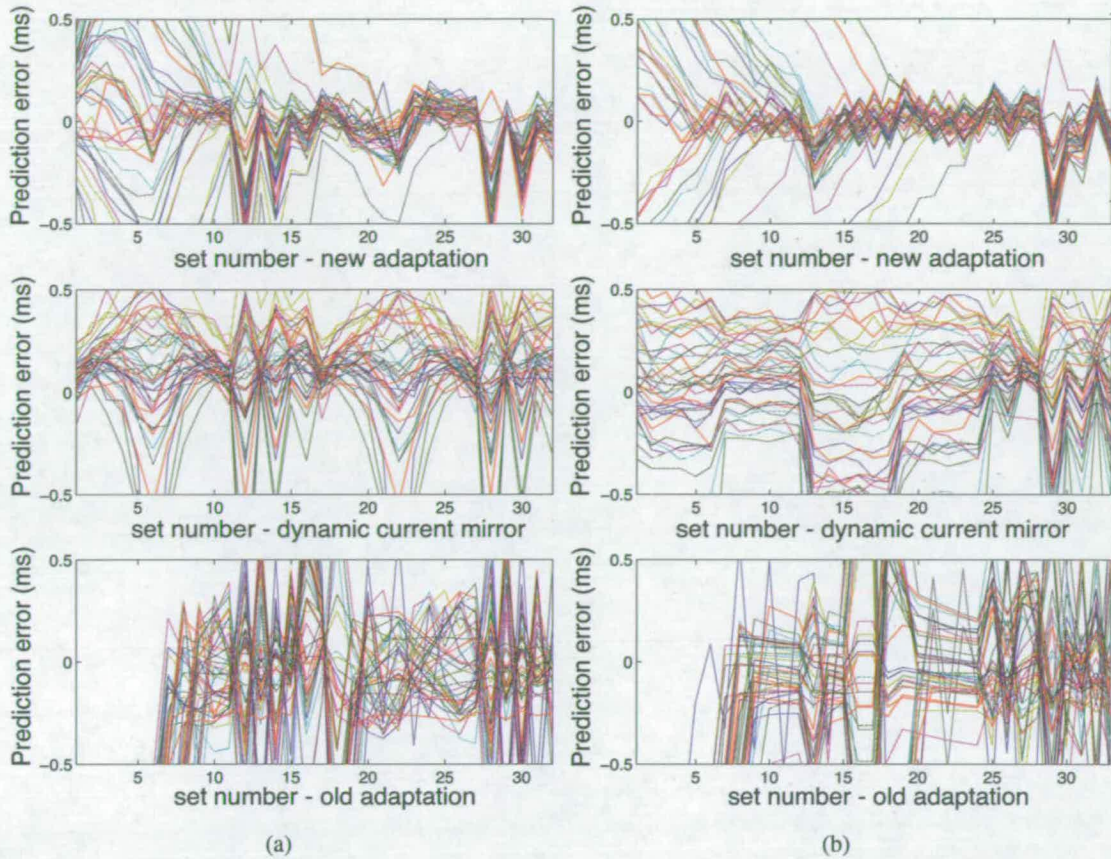


Figure 6.11: *Prediction error during adaptation with varying inter-spike interval, a comparison between different prediction methods. The results of chip tests on the new prediction/adaptation circuit, dynamic current mirror circuit and simulation results for the original prediction/adaptation circuit are compared. (a) a spike train with 16 spike sets repeated twice. (b) a spike train with 33 spike sets.*

due to the appearance of $t(2)$ twice in the prediction calculation. Even if there are no noisy spikes in the interval $t1 - 2$ the prediction may not be made. This situation occurs if a spike 1, spike 2 sequence occurs outside of a prediction and therefore starts one. A new prediction cannot be initiated until the prediction spike has been fired, which may be after the genuine spike 1. A final effect of the noise is to cause unintended adaptation. If a noisy spike 3 happens within the adaptation window of a prediction spike, the circuitry can adapt away from the correct bias point.

After generation, the noisy spike trains were evaluated to see how much the noise would effect a perfect prediction circuit. The worst affected block was block 2 which could only predict 15 out of the 33 times. Block one fared better with 22 out of 33, and block three could predict 25 out of 33 times.

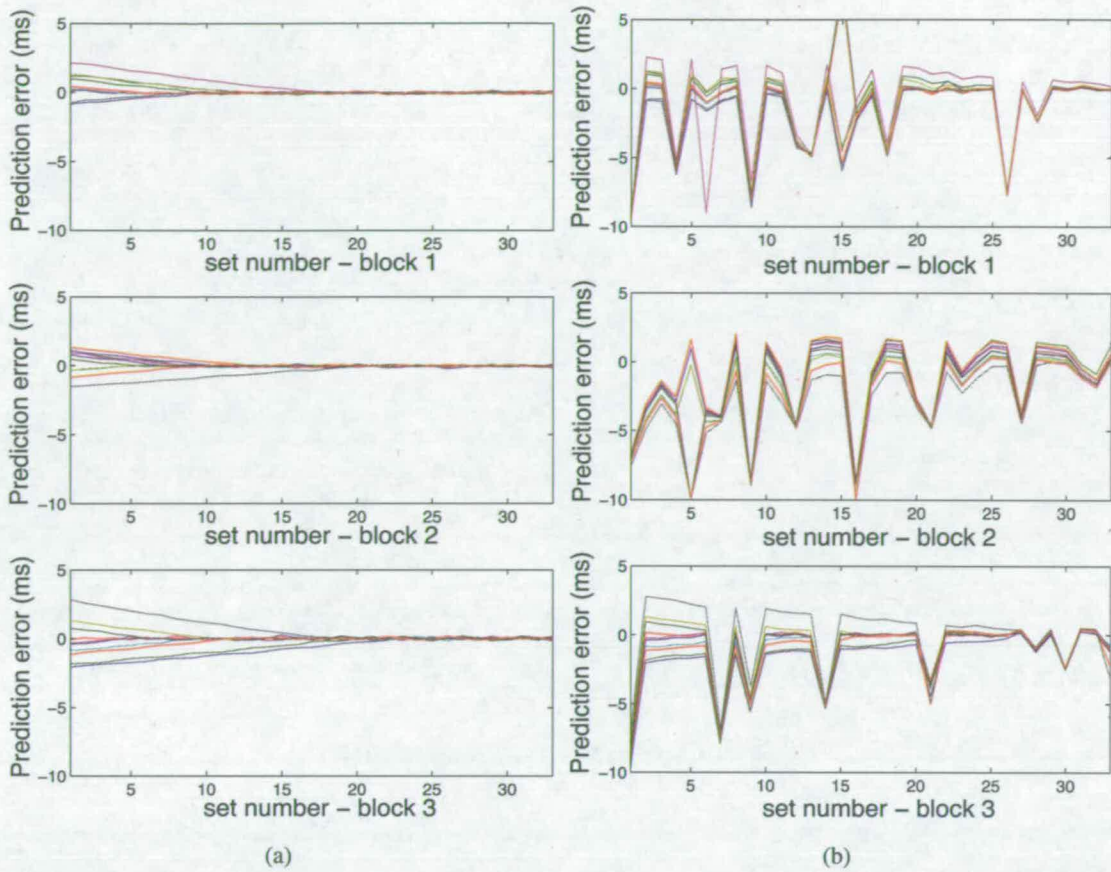


Figure 6.12: *Prediction error during adaptation and the effect of noisy spikes.* 33 spikes sets were applied to ten different chips and the prediction error was recorded. (a) shows the prediction error when no noise is present. (b) shows the prediction error after the addition of noisy spikes. The effect of the adaptation is no longer clear.

To test the ability of the circuit to cope with noisy spikes, the noisy spike pattern was applied as input to ten chips. The prediction error was calculated as the time between the actual spike and the closest prediction. The results are shown in figure 6.12 beside results for a noise free test. The results for each prediction block are shown separately and each trace is the result from a different chip.

From the results in figure 6.12(b) it is difficult to establish whether the adaptive process has continued to operate correctly or not. The prediction error no longer converges in an obvious fashion towards zero and large prediction errors are visible. In block three where the effective noise level was quite small, allowing most of the predictions to be made, some of the adaptive process can be seen. Some of the mismatch between neurons is also highlighted by these results.

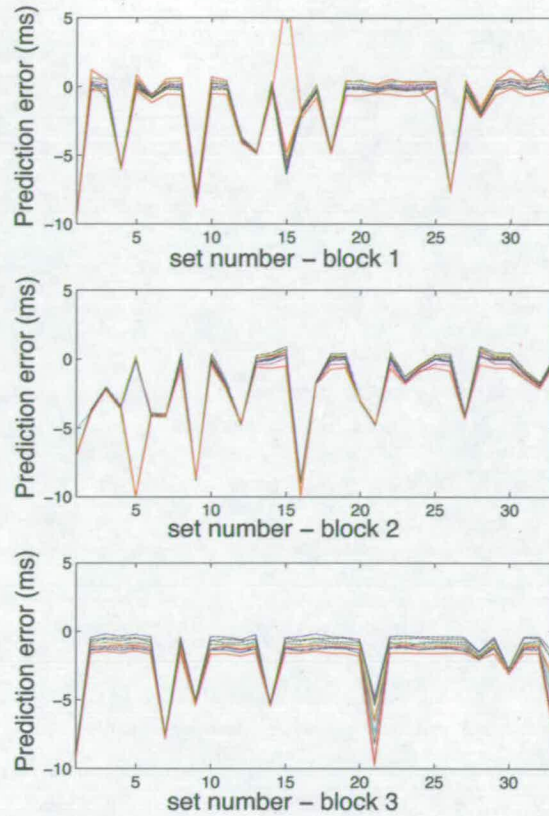


Figure 6.13: Prediction error and the effect of noisy spikes to the dynamic current mirror circuitry. As the prediction error is fairly constant when a correct prediction can be made the timings of very different prediction errors indicate where noise affected the ability to make a prediction. Note that for block three the base line error is approximately 1ms as the third block had a 1:1.2 prediction ratio.

The single wildly inaccurate prediction in set 6, block 1 and the split in predictions at set 5, block 2 are both due to some neurons being slower to fire than others, which resulted in a noisy prediction not being finished when the correct prediction was due to start. The split at set 15, block 1 occurred as an incorrect prediction was made either side of the actual spike at roughly equal intervals. As the closest was taken, it sometimes gave a positive error and sometimes a negative one.

The noisy spikes were also applied to the dynamic current mirror circuitry and the results are shown in figure 6.13. As the performance is fairly constant when making a prediction is possible, it is easier to see when noise prevented an accurate prediction being made.

This information was used to remove the predictions which could not possibly be accurate from figure 6.12(b), and the remaining predictions are shown in figure 6.14.

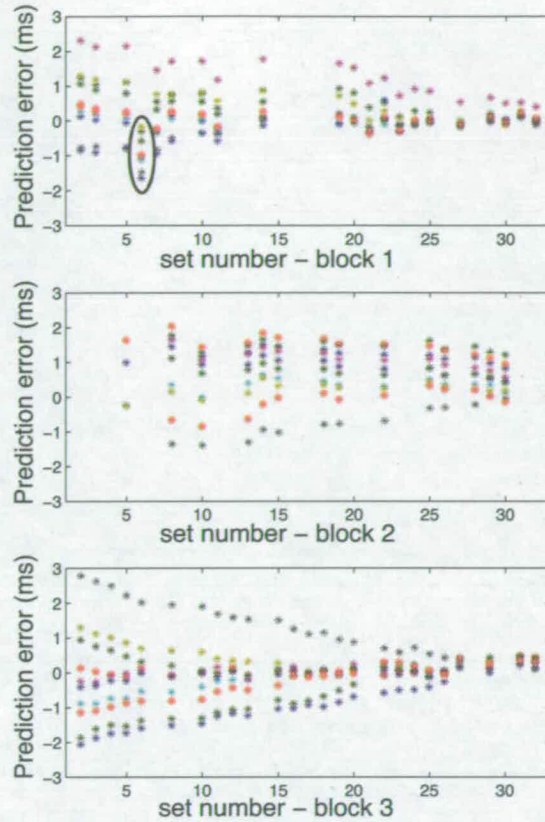


Figure 6.14: *Prediction error during adaptation while noisy spikes are applied. The prediction errors, where it was impossible to make a correct prediction, have been removed. The fact that the adaptive process continued despite the noise is now easily seen. The circled results show a point where noise caused the circuitry to adapt away from the correct bias point.*

From this it is much clearer that the adaptation does indeed continue despite the noise. The adaptation is slowest when the noise level was highest but this is to be expected. The circled results are a good example of when a noisy spike 3 can cause the adaptation to move away from the desired point, however the prediction had recovered by the next set.

6.6 Summary and Conclusions

The circuitry described in chapter 5 suffered from the flaw that the adaptation corrected for a specific error, rather than actually adjusting a current until both the charging and discharging currents matched. The circuitry in this section were designed to rectify that.

The prediction circuitry was modified slightly, to allow the bias voltage for the discharging

current to be directly connected to the charging bias voltage, or to be set by the voltage across a capacitor. The voltage across the capacitor could then be increased to reduce the discharge time/increase the discharging current, or decreased to increase the discharge time/decrease the discharging current.

A switched capacitor scheme was then designed which could increase or decrease the voltage across the capacitor through charge sharing. As the transistor providing the discharging current was operating subthreshold, the charge change across the capacitor had to be very small, resulting in small capacitors sharing charge with the large capacitor storing the bias voltage.

A spike timing dependent algorithm was used to specify the amount of charge that would be shared and whether the voltage across the smaller capacitors would be greater or smaller than the voltage currently across the bias capacitor.

A dynamic current mirror prediction circuit was also fabricated to allow a comparison with an alternative method of prediction.

The chip was tested by applying both spike sets with static and inter-spike intervals. When the static spike sets were applied, the circuit performed better than both the previous adaptive circuit design and the dynamic current mirror prediction circuitry. The standard deviation of the error was reduced to less than 0.1ms for a 5ms inter-spike interval. The circuitry performed best when the inter-spike intervals were varied as well. On the second presentation of a varying inter-spike interval pattern, the improvement which the adaptation had introduced could be seen.

Finally noisy spikes were added to the input spike train, to evaluate the adaptive circuitry's robustness to noise. The level of noise affected the speed of adaptation whenever the noise caused either adaptation away from the desired bias point, or the prediction error to be outside of the adaptive window, leading to no adaptation. The circuitry did continue to adapt, though, whenever a correct prediction was made, adding validity to the assumption that if more correct predictions could be made than incorrect ones, the circuitry would work.

The new circuitry is not perfect. The required changes in bias voltage were very small. In order to achieve this a very large capacitor was used to divide down the increment/decrement voltage. The other smaller capacitors used in the switched capacitor technique were minimum size with their capacitance being of the order of the capacitances of the track and gates connected to it.

This meant that not only was the circuitry large, but fine control of the adaptation was limited by parasitic capacitance. It would be advantageous to be able both to reduce the size of the larger capacitor, and at the same time increase the size of the others. In order to do this the circuitry generating the voltage above and below the current bias voltage would have to be able to give out a smaller voltage differential.

While not all the problems with the initial circuit design have been dealt with, this circuitry provided reliable, repeatable results with a significant improvement on the previous design. It also re-emphasised that spike-timing adaptation could be used to minimise the effect of transistor mismatch.

Chapter 7

Summary and Conclusions

The aim of this work was to examine whether spike timing dependent plasticity can be used to minimise the effect of mismatch within a neuromorphic system.

7.1 Work Undertaken

A depth-from motion algorithm was used as an exemplar for the neuromorphic system which, when fabricated, would have its performance degraded by transistor mismatch. Chapter 2 provided an introduction to the depth-from-motion algorithm as well as reviewing previously published literature on spiking neurons, temporal codes, spike timing dependent plasticity, neuromorphic circuitry and transistor mismatch. Chapter 3 then described the peripheral circuitry needed to implement the prediction element of the depth-from-motion algorithm.

Chapter 4 described three potential methods of prediction. The first relied on capacitive rather than transistor matching. The second employed a dynamic current mirror. The third had a similar configuration to the dynamic current mirror but did not use dynamic biasing. The first two methods were tested using Monte Carlo simulation to establish the minimum level of performance the third had to achieve after spike timing dependent adaptation was applied.

Chapter 5 described the first method of adaptation applied. The time difference between the predicted and actual spike was used to update a weight voltage. This weight voltage controlled an offset applied to the prediction. It was shown that that an offset could be learned that improved the prediction. The performance improvement was significant, reducing the standard deviation of the error from 1.23ms to 0.11ms for a 5ms prediction. It was also shown that when the neurons were activated using output from a transient detecting pixel, the performance was not affected.

While the performance improved greatly, this adaptation method could only apply an offset correction, and as the error in the prediction was related to the current mismatch and the time

to be predicted, determined by the depth of the edge, this method was not capable of providing an accurate prediction for different inter-spike intervals. The adaptive circuitry described in chapter 6 was designed to address this issue. The new design altered a bias voltage until the prediction error reduced. This improved the accuracy of the predictions while the inter-spike interval changed, as well as improving the prediction when a static interval was applied. Not only was the standard deviation of the error reduced to 0.06ms for a 5ms prediction, but the magnitude of the largest error was significantly reduced. In addition, with this design, the ability to predict an interval related to, but not identical to, the input delay was tested. The width of two transistors in the predictive circuit was altered so that the prediction was 1.2 times the length of the input interval. It was shown that this did not alter the accuracy of the prediction after adaptation. Results were also presented for a fabricated dynamic current mirror circuit. The prediction with spike timing adaptation outperforms it. The dynamic current mirror circuitry gave a consistent prediction error but it was offset from zero by on average $100\mu\text{s}$. Finally it was shown that additional noisy spikes did not affect the adaptive process as long as there were sufficient correct input patterns.

7.2 Future Work

There are two main areas on which future work should be focused. The first is area reduction. While this adaptive circuitry does significantly reduce the effect of mismatch, it is a significant area overhead. It could be argued that if there is area available to fabricate an adaptive circuit, the area of mismatch sensitive circuits could just be increased, reducing the level of mismatch that way. This may not be possible due to other design constraints [13], and as minimum process dimensions reduce, there may be a maximum level of matching achievable [107] by increasing the transistor area. Even if the only way to improve performance is to provide an adaptive mismatch compensation scheme, the smaller it is the better. The largest single component in the second adaptive scheme was C_{slope} , see figure 6.1. This capacitor was large to enable the weight increment/decrement to be kept small. In [123] a circuit is described that can apply increments/decrements of $50\mu\text{V}$ to a 0.5pF capacitor. This is a better resolution increment, using a capacitor a tenth of the size. This would be a significant saving in one part of the circuit, and it should be possible to design a smaller circuit with equal or better performance now that the concept has been proven.

The second area to be looked at is additional applications. The depth-from-motion algorithm

was very specific, in that mismatch caused an error in timing between two spikes and there was a bias voltage that could be altered to compensate. In [81] STDP is used to strengthen connections from neurons that fire quickly to those with a slower response, which in turn causes the slower neurons to fire faster. The timing difference is caused by mismatch, so this application does reduce the effect of mismatch within the system. It would be interesting to investigate whether this sort of adaptive circuitry could be used outwith the traditional neuromorphic arena in, for example, the matching of currents in a current steering Digital to Analogue Converter.

7.3 Conclusions

This thesis set out to examine whether spike timing dependent adaptation could be used to reduce the effect of mismatch within a neuromorphic system. Through the fabrication of two adaptive circuits it has been shown that it is possible. Both variants of the circuitry allowed the adaptation to continue after the prediction had been improved. This allows for more flexibility than an initial calibration would provide, as should the environmental conditions change, the circuitry can adapt to compensate. The main concern with this method of mismatch compensation is the size of the adaptive circuitry. For this to be a realistic solution, work would have to be undertaken to minimise the adaptive circuitry. As transistor dimensions reduce and mismatch becomes a greater problem, the size of the adaptive circuitry will shrink, making it a more viable solution.

Spike timing dependent adaptation has been shown to be a powerful neural algorithm in simulation. This work has shown that it is also possible to use it to minimise the effect of transistor mismatch. While this work may not be directly continued, it does show that biologically inspired circuits can be applied to real world engineering problems. Spike timing dependent plasticity will also continue to be of interest to neuroscientists. More will be discovered concerning the mechanisms involved in STDP and the problems the neural system applies it to. Engineers should continue to look for other real world applications that can benefit from biologically inspired computation.

Appendix A

Publications

Journal papers

- ***K.L. Cameron**, V. Boonsobhak, A.F. Murray, D. Renshaw; *Spike Timing Dependent Plasticity (STDP) can Ameliorate Process Variations in Neuromorphic VLSI*, IEEE Transactions on Neural Networks, vol. 16, no. 6, November 2005, pp 1626-1637
- Z. Yang, A.F. Murray, F. Worgotter, **K.L. Cameron**, V. Boonsobhak; *A Neuromorphic Depth-from-Motion Vision Model with STDP Adaptation*, IEEE Transactions on Neural Networks, vol. 17, no. 2, March 2006, pp 482-495
- Z. Yang, **K.L. Cameron**, A.F. Murray, V.Boonsobhak; *An Adaptive Visual Neuronal Model Implementing Competitive Temporally Asymmetric Hebbian Learning*, International Journal of Neural Systems, vol.16, no. 3, June 2006, pp 151-162

Refereed Conference Paper

- ***K.L. Cameron**, A.F. Murray; *Can Spike Timing Dependent Plasticity Compensate for Process Mismatch in Neuromorphic Analogue VLSI?*, 2004 IEEE International Symposium on Circuits and Systems (ISCAS 2004), V, 748-751, 2004
- ***V. Boonsobhak**, **K.L. Cameron**, D. Renshaw, A.F. Murray; *Compensating Mismatch in a Dedicated Pixel Array for Moving Edge Detection*, IEEE TENCON Conference on Analog and Digital Techniques in Electrical Engineering, D, 278-281, November 2004
- ***K.L. Cameron**, A.F. Murray, S. Collins; *Spike Timing Dependent Adaptation for Mismatch Compensation*, 2006 IEEE International Symposium on Circuits and Systems (ISCAS 2006), 1223-1226, 2006

Abstracts

- Z. Yang, **K.L. Cameron**, V. Boonsobhak, A.F. Murray; *A Cognitive Vision Model with STDP Adaptation*, 7th International Conference on Cognitive and Neural Systems , 73, May 2004

Those marked with a * are directly related to this work. The others are the result of the EPSRC funded project of which this work formed a part.

Spike Timing Dependent Plasticity (STDP) can Ameliorate Process Variations in Neuromorphic VLSI

Katherine Cameron, *Student Member, IEEE*, Vasin Boonsobhak, *Student Member, IEEE*, Alan Murray, *Senior Member, IEEE*, and David Renshaw

Abstract—A transient-detecting very large scale integration (VLSI) pixel is described, suitable for use in a visual-processing, depth-recovery algorithm based upon spike timing. A small array of pixels is coupled to an adaptive system, based upon spike timing dependent plasticity (STDP), that aims to reduce the effect of VLSI process variations on the algorithm's performance. Results from 0.35 μm CMOS temporal differentiating pixels and STDP circuits show that the system is capable of adapting to substantially reduce the effects of process variations without interrupting the algorithm's natural processes. The concept is generic to all spike timing driven processing algorithms in a VLSI.

Index Terms—Active pixel, CMOS integrated circuits, focal-plane sensor, neuromorphic analogue very large scale integration (VLSI), spike timing dependent plasticity (STDP), temporal processing, transistor mismatch.

I. INTRODUCTION

PELGROM's law states that device mismatch is proportional to $1/\sqrt{WL}$ [1], indicating that the problems associated with mismatch will only increase as device sizes approach submicron geometries. When subthreshold currents are required, for power saving or time constant considerations, mismatches have an even greater effect [2]. Some compensation mechanisms have been proposed, which allow the fabrication of, for example, current mirrors, dividers and comparators [3]–[5] with improved matching characteristics while still using minimum size transistors. We introduce an alternative approach that uses the change in the temporal characteristics introduced by the mismatch to effect a correction.

In most systems, a change in transistor parameters has an effect on the system time constants and mismatch affects the temporal characteristics of the circuit. spike timing dependent plasticity (STDP) is a class of neural algorithms driven not by spike-rate correlations, but by individual interspike timings. This class of algorithms has been shown to recover different and potentially richer information from a spike train when compared to rate and population coding methods [6] and has been shown capable of adapting synaptic delays [7]–[9] to improve network performance. STDP is, however, a relatively simple algorithm which has been successfully implemented in analogue very large scale integration (aVLSI) [10]–[17]. If a

suitable spiking neural system could be implemented in aVLSI the ability to adapt for delay error could be used to correct timing errors introduced by process mismatch.

Our exemplar system was developed by Wörgötter *et al.* [18]. It is a spike timing driven algorithm for visual scene analysis whose temporal characteristics will be degraded if implemented in analogue focal-plane VLSI. Although the algorithm is intrinsically interesting, we use it primarily as a background to test the ability of STDP to correct for mismatch. Brief details of the algorithm and its small-scale implementation in aVLSI circuitry are presented in Section II. The transient-detecting pixel circuitry is described in Section III. Section IV presents the STDP circuit and Section V brings pixels and STDP adaptation together. All circuitry is implemented using the AMS 0.35 μm CSI CMOS process.

II. VISION ALGORITHM BASED ON SPIKING NEURONS

This section describes a parallel noise-robust algorithm [18] operating on the neurons of an artificial retina to recover depth information from radial flow fields in a visual scene. This algorithm finds its motivation in the behavior of animals. In different species, varying strategies are observed in order to reduce the optical flow to, if possible, one-dimensional motion. In particular, for airborne animals such as flies and birds, this actually leads to the tendency to fly in straight lines. Ideally, it means that only forward motion exists and that optical flow is reduced to its radial components. Under these circumstances, points in the image plane move radially outwards from the focus of expansion with a velocity proportional to radial distance from the focus of expansion but inversely proportional to their depth in the image. Hence, the depth can be inferred from this image expansion.

The retina in this algorithm, shown in Fig. 1, consists of radially arranged neurons connected only to their nearest neighbors in both directions on the same radius. It functions as a visual system of a moving robot. The robot is constrained to move along the optical axis of the retina. As the robot moves, objects in the field of view are projected on the retinal plane, and neurons are excited as soon as a sufficiently strong brightness transient, i.e., a moving edge, occurs. The time difference between two subsequently activated neurons (e.g., neurons 1 and 2) is used to compute the depth information. The structure of the network is such that all computations remain local, i.e., neurons connect to nearest neighbors, which facilitates parallelization. However, the local nature of all calculations makes the algorithm sensitive to noise, for example that caused by camera jitter. A prediction mechanism is introduced to reduce this sensitivity.

Manuscript received October 1, 2004; revised February 1, 2005. This work was part of a joint EPSRC project involving the Universities of Edinburgh, Oxford and Stirling. Vasin Boonsobhak's work was supported by Mahanakorn University of Technology, Thailand.

The authors are with the School of Engineering and Electronics, The University of Edinburgh, U.K. (e-mail: K.L.Cameron@sms.ed.ac.uk).

Digital Object Identifier 10.1109/TNN.2005.852238

The calculated depth is used to predict when the last-excited neuron (e.g., neuron 3) will fire. Events are accepted as valid if this prediction corresponds with an actual event.

We have based our pixel upon the light-transducing elements described in [19] and [20]. The technique for device mismatch correction is based upon a form of asymmetric Hebbian learning, a neural algorithm implementing STDP.

The system block diagram is shown in Fig. 2. The spike firing circuit comprises three leaky integrate-and-fire (LIF) neurons [13]. These LIF neurons integrate output currents received from transient-detecting pixels (E), cancel the dark current, and fire when the accumulated signal reaches a threshold. The spikes from neurons 1 and 2 are passed to the prediction network which generates another spike, spike 3p (i.e., “predicted,” not actual), at a time determined by the firing times of the input spikes. The timing of spike 3p can then be compared to that of spike 3 by the spike confirmation block. In the absence of process variations, Spikes 3 and 3p will be coincident. Spike 3 is, therefore, accepted as genuine only if it arrives within the time window of spike 3p that represents the acceptable tolerance on the prediction. The adaptive STDP network is also shown. It aims to “adapt out” the process-induced mismatch between the actual firing time $t(3)$ and its predicted time $t(3p)$.

III. TRANSIENT-DETECTING PIXEL

In this section, we present a compact integrated circuit suitable for use in a focal-plane image processing pixel. Its main function is the enhancement of local, temporal brightness transients.

A. Circuit Description

The pixel circuit is shown in Fig. 3. It combines an adaptive photoreceptor with a rectifying differentiating element and consists of a photodiode D in series with a transistor M_n in source-follower configuration. A negative feedback loop is created between the source and the gate of M_n . The feedback loop consists of a high-gain inverting amplifier in common-source configuration (M_n , M_p), a rectifying temporal class-AB differentiator stage (M_{ON} , M_{OFF} , M_{AB} , $C = 500$ fF) and a capacitive gain stage ($C_1 = 1$ pF, $C_2 = 50$ fF). We employed Nwell-over-Psubstrate photodiodes as opposed to Ndiff-over-Psubstrate photodiodes used in [19] and [20]. This increases the pixel circuit response to illumination changes because the parasitic capacitance is reduced by a factor of about 10 [21]. The capacitors occupy a large part of the pixel area. Hence, for a $20\ \mu\text{m} \times 20\ \mu\text{m}$ photodiode in our pixel circuit, the fill factor is about 10%.

The class-AB differentiator brings two benefits. First, it eliminates the “dead” band inherent in its class-B counterpart [20], which can slow down the response when a voltage signal is present within the band. Second, it provides some current to counterbalance leakage currents at the differentiator node, reducing asymmetry between positive and negative transients.

An additional capacitive gain stage in the feedback loop gives enhancement to the response [19]. The voltage variations on the differentiator node V_{diff} are amplified with respect to the variations of V_n by the capacitive divider ratio $A_C = (C_1 +$

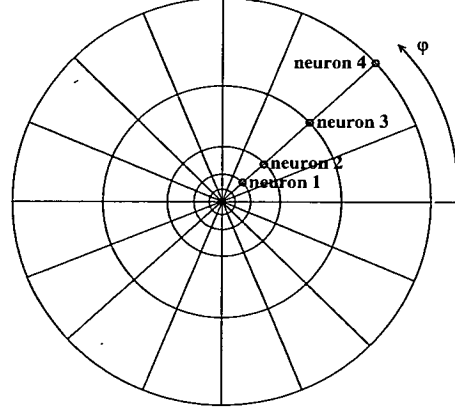


Fig. 1. Layout of the retina. The radial arrangement of the neurons allows depth information to be calculated as described in [18].

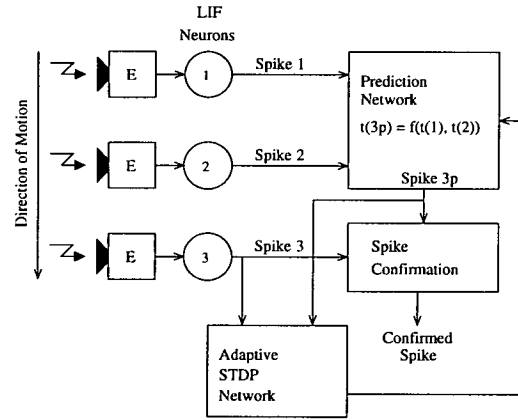


Fig. 2. System block diagram. Neurons 1 to 3 fire when an edge passes the corresponding pixels. Neuron 3p fires at a time predicted from the firing time of neurons 1 and 2. If the neurons fire at equal intervals, $t(3p) = t(2) + \{t(2) - t(1)\}$ where $t(1)$ is the firing time of spike 1, $t(2)$ for spike 2 etc. Spike 3p is the prediction of spike 3.

$C_2)/C_2 = 21$ and so are the transient currents I_{ON} and I_{OFF} . The resistive element, constructed from two oppositely directed diodes in series, ensures that V_{diff} eventually adapts to a dc value close to V_n . In ideal circumstances, the current is limited to a low value by the reverse-biased diode, such that the element exhibits a symmetric, saturating, sigmoidal current-voltage characteristic.

A full circuit analysis of the steady state and transient responses of the pixel is presented in the Appendix.

B. Experimental Setup

The power supply voltage V_{dd} was set to 3.3 V and the bias voltages V_{b1} , V_{b2} , and V_{b3} were fixed for all measurements such that M_p and M_n were operated slightly above threshold. As the pixel provides output in the form of current ($I_{ON} + I_{OFF}$),

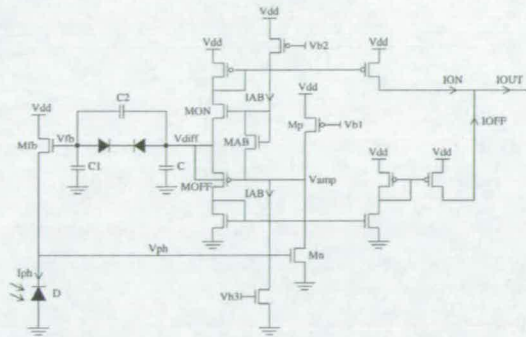


Fig. 3. Circuit diagram of the transient-detecting pixel. Positive or ON transients and negative or OFF transients, although they can be assigned to separate terminals, are combined on a single output node and fed to a neuron. The circuit has a class-AB differentiator and a capacitive gain stage, which eliminates the "dead" band that can slow down the response and amplifies changes of the sustained voltage, respectively.

a 1 M Ω resistor was connected between the output node and ground, enabling voltage readout for an oscilloscope. As a result, the output time constant was massively increased and does not reflect the natural on-chip response of the pixel.

The entire chip was illuminated through a diffuser with a radiance-modulated light-emitting diode (LED), operating around an emission wavelength of 590 nm, while shielding the chip from ambient light (Fig. 4). A forward voltage was applied to the LED with a series resistor, such that the current and, thus, the radiance of the LED were approximately linear with the applied voltage in a certain range. A radiance-voltage calibration was performed using a digital light meter.

C. Test Results

For steady-state measurements, the irradiance range was expanded by inserting neutral density (ND) filters with attenuation factors of 10, 100, and 1000 between the diffuser and the chip. The voltage V_{fb} at the feedback node of the transient-detecting pixel shows the predicted logarithmic behavior over a large irradiance range, as shown in Fig. 5. The dark line amidst the data points shows a linear-logarithmic relationship between V_{fb} and illuminance.

For transient measurements, the LED was modulated using a square wave signal from a waveform generator which was buffered and low-pass filtered. The low-pass filtering was necessary to reduce artifacts due to the discretization of the waveform generator's output signal. The time constant of the low-pass filter was set to 1 ms. The circuit was allowed to reach a steady state at a default illuminance of 1 lux. Saturation and adaptation of the pixel can be observed in the V_{amp} and V_{diff} traces for a contrast step of 100 lux, as shown in Fig. 6. Since the transient currents I_{ON} and I_{OFF} of the pixel circuit are very small (a few hundred nA), a large irradiance step (100:1) is necessary in order to display a reasonably large V_{out} waveform (a few hundred mV) on the oscilloscope. However, when testing the combined pixel/STDP circuit as will be described in Section V, we used a smaller irradiance step (15:1).

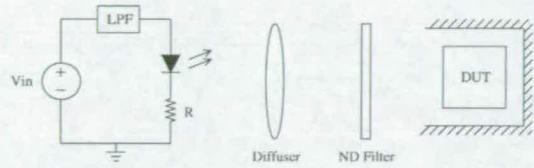


Fig. 4. Transient-detecting pixel's test setup. A diffuser is used to produce uniform illumination over the entire chip. ND filters with different attenuation factors expand the irradiance range for steady-state measurements.

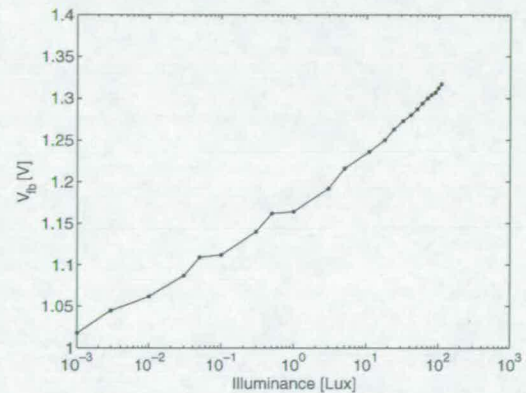


Fig. 5. Steady-state characteristic of the transient-detecting pixel. The logarithmic irradiance is encoded in the voltage V_{fb} at the feedback node over an irradiance range of five decades.

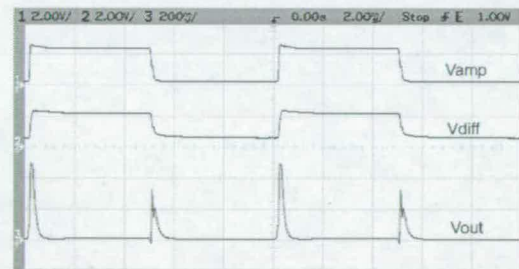


Fig. 6. Irradiance step response of the transient-detecting pixel. These oscilloscope traces show V_{amp} , V_{diff} and $V_{out} = 1 \text{ M}\Omega \times (I_{ON} + I_{OFF})$. The increase in the V_{amp} and V_{diff} traces is due to a positive contrast step of 100 lux with the near-ground baselines represent a default illuminance of 1 lux. The V_{out} trace shows the temporal derivative of the input contrast step.

IV. SPIKE TIMING DRIVEN CIRCUITS

In these concept-proving experiments, the output from three transient-detecting pixels drives three LIF neurons. As an edge passes a pixel the corresponding neuron fires a spike. These spikes are used directly by the STDP adaptive circuit. The two main components making up the spike timing driven circuit are the predictive and the adaptive circuits. The adaptive circuit is designed to minimize the effects of process mismatch in the prediction.

The results described in this section are from a replicated copy of the circuitry with voltage controlled inputs acting as artificial pixels. This allowed initial testing of the spiking circuitry in isolation from the pixels.

A. Prediction Circuitry

Wörgötter *et al.* [18] showed that knowledge of a pixel array layout can be used to infer when an edge will pass a particular pixel from the time of previous edges. Three pixels suffice to demonstrate the success of our approach and their layout was chosen such that the time between the first and second spikes is equal to the time between the second and third. Therefore the predicted time of arrival for the third spike, spike 3p, is $t(3p) = t(2) + [t(2) - t(1)]$.

The prediction spike is generated by comparing a voltage across a capacitor to a reference voltage as shown in Fig. 7. The charging and discharging currents connected to C1 are designed to be identical so that the time to discharge the capacitor is equal to the charging time. The sequence of events is as follows:

- $t(1)$ Spike 1 is fired and P6 and N6 act as a switch which sets the initial value of V_{ramp} to V_{rampth} .
- $t(1)-t(2)$ P5 is on and C1 is charged through P3 and P4 which supplies a mirrored version of the current set by V_b . P5 turns off while N5 becomes active.
- $t(2)$ P5 turns off while N5 becomes active.
- $t(2)-t(3p)$ C1 is discharged through N3 and N4. If the current source and sink are matched the time taken to discharge C1 should match the charging time.
- $t(3p)$ V_{ramp} is compared to V_{rampth} using a differential pair. When V_{ramp} crosses it, neuron 3p is "fired." This ends the discharge period.

The control signals $t1-2$ and $t2-3p$ are generated by spike triggered SR Latches, e.g., set connected to spike 2 and reset connected to spike 1 results in $t1-2$.

The accuracy of the prediction depends upon how well the charging and discharging currents are matched. We are primarily interested in gain mismatch and therefore transistors P1, P4, N2, and N4 are used to minimize the current change caused by the early effect. Fig. 8(a) shows the result if the currents are not matched. First, the charging current is smaller than the discharging current and then the situation is reversed. The currents will not be perfectly matched unless post-fabrication trimming is performed on the circuit. This is both costly and time consuming. Instead, we use the relationship between the actual and predicted third spike to correct for any mismatch.

It should also be noticed that the addition of a second power supply pin, V_{dd_mirr} , to the circuit in Fig. 7 allows mismatch to be forced on to the circuit. This facilitates testing of the adaptive network under a wide range of induced prediction errors, which would not be possible with the limited number of chips received. Under most test conditions the two pins are shorted together.

In future chips with larger pixel arrays the prediction for different layout configurations can be achieved by ratioing the charging and discharging currents to alter the time between spikes 2 and 3p.

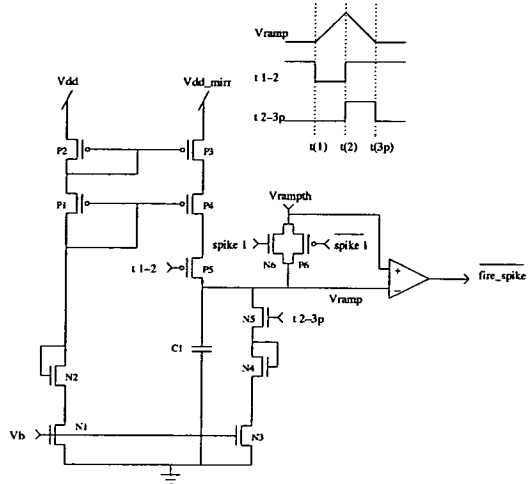


Fig. 7. Spike prediction circuit. C1 charges between $t(1)$ and $t(2)$ and discharges from $t(2)$ to $t(3p)$. The charging and discharging currents are matched to enable prediction of spike 3 at the time when C1 has returned to V_{rampth} .

B. Adaptive Circuitry

To correct the prediction error a method must be found either to adjust one, or both, of the currents until they are matched or to compensate for the effect of the difference in the currents. Changing the voltage across C1 at the time of the switching between charging and discharging, $t(2)$, will alter the discharge time. The voltage can be increased or decreased, which has a direct effect on the discharge time and therefore compensates for the current difference. This method was chosen and examples are shown in Fig. 8(b).

The change in the voltage across C1 is achieved using the excitatory/inhibitory synapse described in [22] and is shown in Fig. 9 along with the output waveform. The circuit is designed such that if V_w is greater than V_{synth} the synapse is excitatory. Should V_w be smaller the synapse is inhibitory. $I_{in/out}$ is only connected to V_{ramp} at $t(2)$ and the value of V_w will determine the amount and the direction that the peak voltage will change. The width of spike 2 and therefore the time $I_{in/out}$ is connected to V_{ramp} will be constant for a given set of environmental conditions. Should these change V_w should change accordingly.

Transistors N1 and N6 are both small W/L transistors cascaded by the wide transistors N2 and N7. This maintains N1 and N6 in the linear region over a wide range as described in [12] and [13]. N1 is biased by V_{synth} into the linear region. The value of V_{cas} determines the minimum value of V_w that N6 enters the linear region and will set the maximum amount of current that can be injected/removed from C1. It also determines the gradient of the $I-V$ curve which has a direct effect on the amount the output current changes with a step change in V_w . The smaller the range of required current the more accurate an output current can be selected.

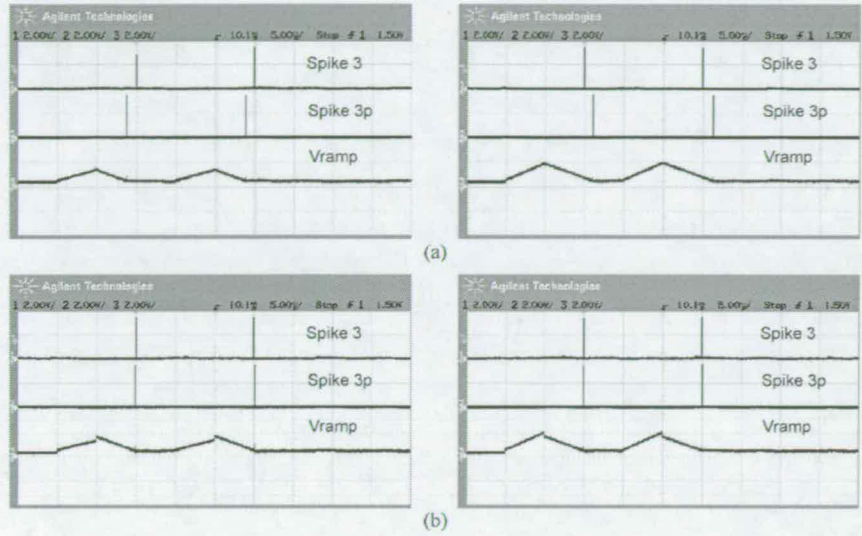


Fig. 8. Pre- and postadaptation waveforms. These oscilloscope traces show the spike 3, spike 3p and Vramp signals during testing. At $t(1)$ Vramp is set to Vrampth and charging begins. At $t(2)$, the current direction is switched to start the discharge. Spike 3p is generated when Vramp returns to Vrampth. The timescale shown is 5 ms/div. (a) Preadaptation and the error in spike 3p is clear. (b) The signals postadaptation where the error has been reduced. The alteration to Vramp can also be seen.

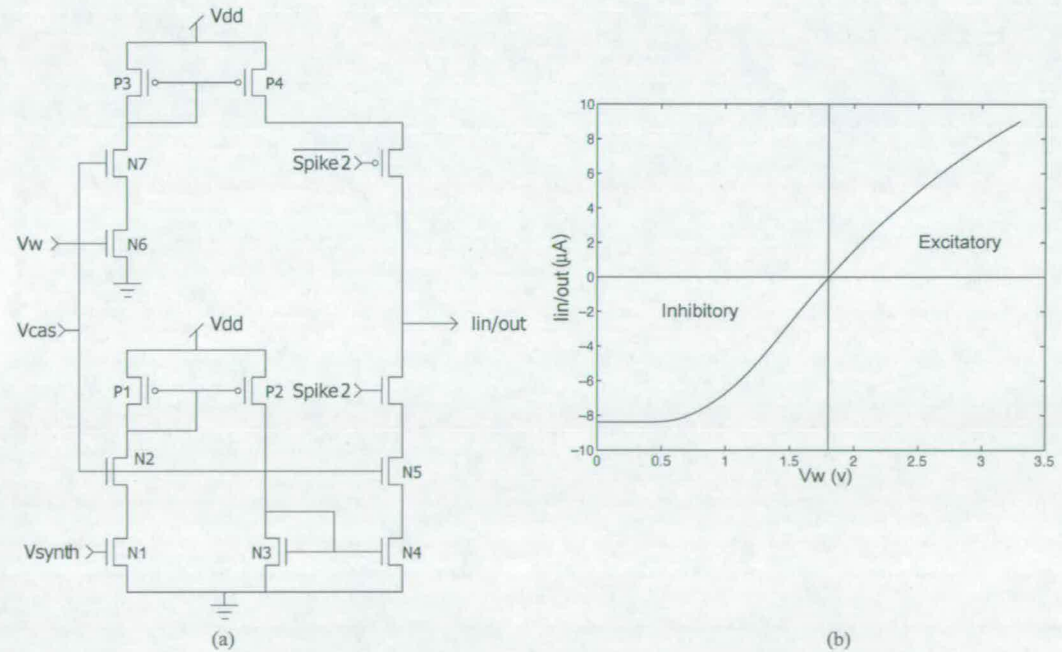


Fig. 9. Excitatory/inhibitory synapse circuit and I - V graph. The synapse is excitatory if V_w is greater than V_{synth} and inhibitory if V_{synth} is greater. (a) Shows the circuit configuration. (b) Is the synapse output current with $V_{synth} = 1.8$ V. $I_{in/out}$ is zero when $V_w = V_{synth}$.

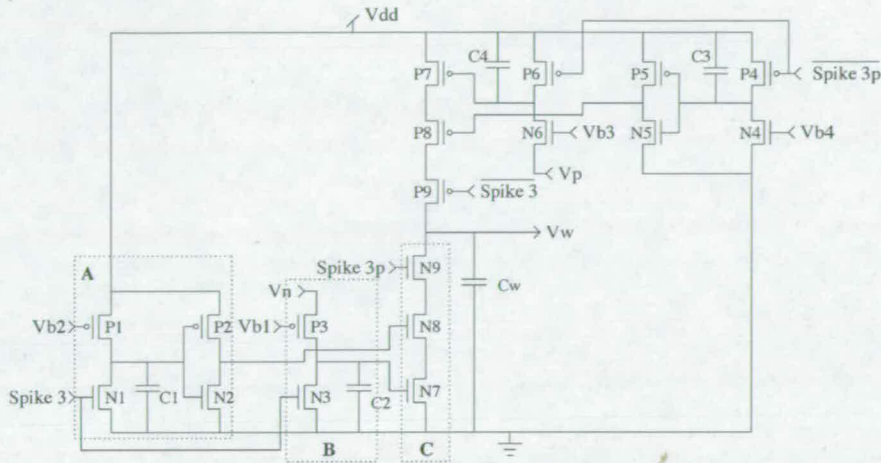


Fig. 10. Weight adaptation circuit. V_w is increased if spike 3p occurs before spike 3 and decreased if neuron 3 fires first. Weight change only occurs if spikes 3 and 3p happen within a specified time window. The weight change (ΔW) is “capped” by V_n and V_p to prevent V_w moving from one supply rail to the other. Three circuit blocks are highlighted. A: The circuit setting the time window. B: The circuit setting the amount of weight change. C: Combining the two together to create the negative part of the ΔW graph.

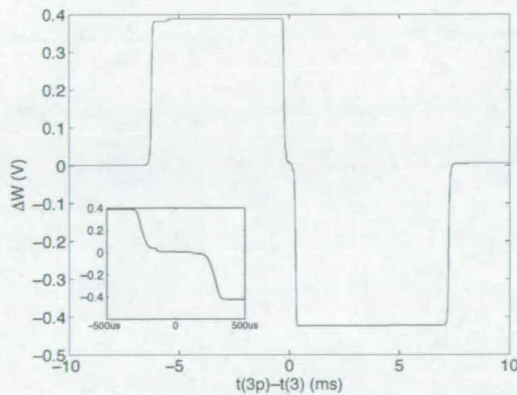


Fig. 11. Weight adaptation algorithm. The weight change algorithm implemented by the circuitry shown in Fig. 10. The change in ΔW was measured in simulation. The inset picture is the range $-500 \mu s - 500 \mu s$. The null point around $t(3p) - t(3)$ occurs when charge injection due to switching dominates.

Most variants of STDP maximize the synaptic weight-change when spikes are near-simultaneous. In contrast, our system is designed to apply weight changes which result in the spikes becoming more coincident.

The setting of V_w must be controlled in order that the appropriate value is set to give coincidence of spikes 3 and 3p. The circuit shown in Fig. 10 provides this control by implementing the weight-change curve shown in Fig. 11.

The circuit in Fig. 10 has two distinct parts. MOSFETs N1-3 and P1-3 control reductions in V_w (depression) through N7-9, while increases (potentiation) are achieved through transistors N4-6 and P4-9. As the potentiation and depression

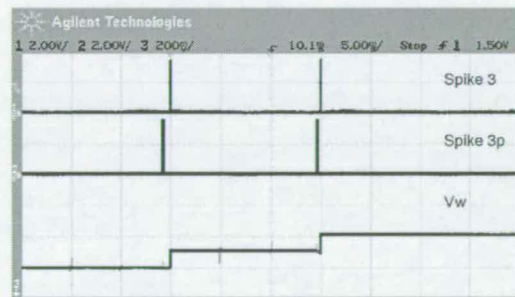


Fig. 12. Weight change. The difference in spike timing alters the synaptic weight and the prediction becomes more accurate. The timescale shown is 5 ms/div.

circuits are mirror-images, with transistor polarities changed, we will present a detailed description of only the depression mechanism.

When spike 3 occurs C1 is discharged through N1 and then charges slowly to V_{dd} through P1. This results in a pulse at the gate of N8, whose width is determined by V_{b2} . This pulse defines the window within which weight change occurs. At $t(3)$ C2 is discharged through N3 and then charged slowly to V_n through P3. This voltage is N7's gate voltage and controls the size of the weight change. As the voltage across C2 increases with time, the amount of weight change will increase the longer the delay between $t(3)$ and $t(3p)$ until the voltage reaches V_n . This defines the maximum weight change. If spike 3p occurs within the time window, C_w is discharged through N7, N8, and N9 by an amount related to the voltage across C2. When the weight voltage is decreased, the peak of V_{ramp} decreases. This reduces the discharge time and spike 3p will occur at a time $t(3p)$ that is closer to $t(3)$.

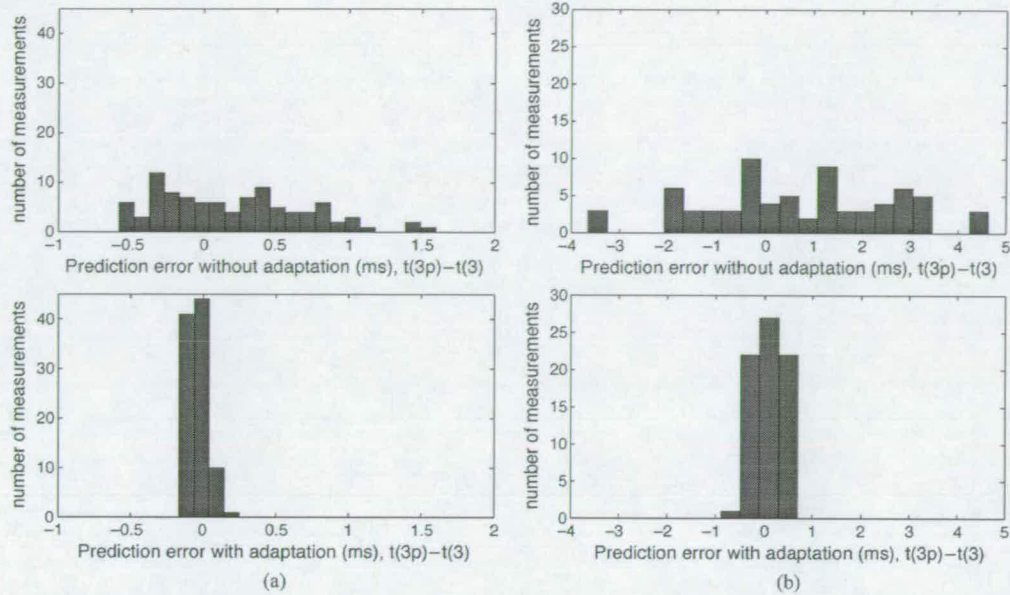


Fig. 13. Distribution of prediction errors. The prediction error is defined as $t(3p) - t(3)$. Each graph has the preadaptation results above the postadaptation ones. The results were obtained with V_{dd_mirr} varying between 3.28 and 3.31 V. Results were when (a) $t(2) - t(1) = 1$ ms and (b) $t(2) - t(1) = 5$ ms.

The weight change is “capped” by V_n and V_p to prevent V_w moving from one supply rail to the other. The window width is determined empirically from *a priori* knowledge as to when two spikes may be regarded as unrelated events.

When choosing the bias voltages V_{b1} , V_{b3} , V_n , and V_p , there is a trade-off to be made between speed of convergence and the width of the null point, determined by V_{b1} and V_{b3} , around $t(3p) - t(3) = 0$ which determines how close spikes must be before the time difference has a negligible effect on the weight. This results in a tolerance associated with the prediction. The maximum weight change set by V_n and V_p has an effect on how many spike pairs are required to correct a particular prediction.

If the adaptation windows, set by V_{b2} and V_{b4} , are too small, large mismatch errors will be judged to be uncorrelated spikes and no adaptation will occur. This leads to a further trade off between the circuit’s ability to adapt and the maximum allowable density of edges in the scene. A larger window will allow a greater range of errors to be corrected but, for the circuitry to function, no two edges can occur within that interval.

When spike $3p$ is a good prediction of spike 3 no further weight change occurs. It is assumed that correct edges will vastly outnumber incorrect edges and that if one does fall within the adaptation time window it will only cause a small deviation of weight which will be corrected by subsequent correct spikes. The adaptation network can therefore be left active after this “self-calibration” and will continue to compensate for any circuit drift caused by changes in environmental conditions or charge leakage from the capacitor. The latter is particularly important as many compensation techniques suffer from problems in the permanent storage of calibration values.

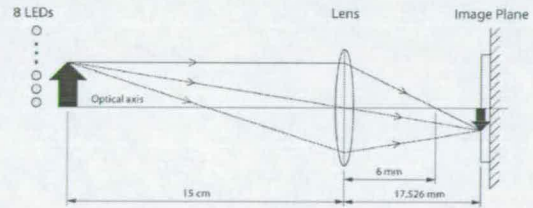


Fig. 14. Combined pixel/STDP circuit’s test setup. Moving patterns from a flashing LED array are projected on the chip surface by a surveillance camera lens.

C. Test Results

When the circuitry was fabricated, using the AMS $0.35 \mu\text{m}$ CSI process, a copy of the predictive and adaptive circuit was laid out with voltage controlled inputs to allow testing in isolation from the pixels. This was also the circuit that was tested with varying V_{dd_mirr} , see Fig. 7.

The oscilloscope traces in Fig. 8 are taken from one of the chips when the input neurons were fired with a 5 ms interneural delay. It can be seen that the STDP adaptation improves the prediction significantly. Fig. 12 shows part of the adaptive process. As the weight increases the resultant improvement in the prediction can be seen.

The circuit was further tested by activating the three input neurons with 1 and 5 ms interneural delays and measuring the prediction error, $t(3p) - t(3)$ before and after adaptation. This was done three times on eight different chips with V_{dd_mirr} varying between 3.28 and 3.31 V. Fig. 13 shows the pre- and

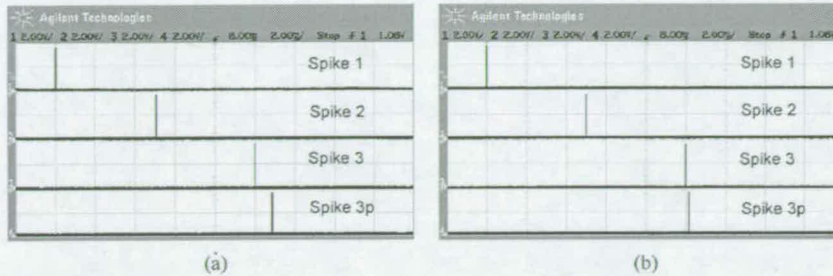


Fig. 15. Pre- and postadaptation results. These results show the spike 1, spike 2, spike 3 and spike 3p measured during experiments when $t(2) - t(1) = 5$ ms. The timescale is 5 ms/div. (a) Is preadaptation and the error in spike 3p is clear. (b) Shows the signals postadaptation where the error has been reduced.

postadaptation error when the interneural delay is (a) 1 ms and (b) 5 ms. These results compare favorably with the simulation results reported in [23].

V. SMALL ADAPTIVE PIXEL ARRAY

The combined pixel/STDP circuit was tested by stimulating it with moving light patterns generated by an array of flashing LEDs (10 lux) as shown in Fig. 14. The LED test patterns were set in such a way that transient-detecting pixels were illuminated chronologically. The optical part of the imaging system was a surveillance camera lens with a focal length of 6 mm and an f -number of 1.2. The LED array was placed at a distance of 15 cm from the lens. This experiments were repeated 10 times over five chips and conducted under the ambient lighting condition of fluorescent office lighting (150 lux).

Fig. 15 shows an example of spike waveforms observed in the experiments before and after STDP adaptation. It is important to note that each pixel is able to detect the LED test pattern (high-frequency signal) while rejecting the ambient light (low-frequency signal).

After the generation of spikes had been verified the prediction delay was measured before and after adaptation. For this test V_{dd_mirr} was kept at 3.3 V and the interneural delay times used were 1 and 5 ms. Fig. 16 shows the resultant spread of the errors. It can be seen that adaptation improves the prediction substantially. When these results are compared with those taken from the voltage controlled circuit when V_{dd_mirr} was 3.3 V, Fig. 17, it can be seen that the pixel does not affect the spread of error. The pixel driven, pre-adaptation results have a less smooth distribution, as these results were taken from 5 chips rather than the 8 for the voltage driven circuit. The mean prediction error and the standard deviation was measured for each distribution and can be found in Table I.

The similarity of the numbers in Table I between the pixel and voltage driven circuitry confirms that pixel activation does not introduce additional error. Small differences should be expected as the results are being collected from a small sample of die.

The error when the interneural time delay is 5 ms will always be greater than the 1 ms error, because the prediction error is related to the mismatch in current and charging time. The mean values shown in Table I can be seen to have a positive bias. Transistors N1 and N3 in Fig. 7 will be affected by the early effect

and this would cause a small positive bias but simulation results indicate it should not be as big as the value seen. We believe the majority of the bias is caused by connecting the drains of the current mirror, V_{dd} and V_{dd_mirr} , externally to the chip rather than with a direct metal connection. A voltage drop of a few mVs on the V_{dd} line would explain this systematic offset. This chip is therefore not suitable for determining the level of mismatch in this process but it does illustrate the effectiveness of an adaptive, STDP approach for correcting mismatch. The adaptation does move the mean back toward zero and the standard deviation is greatly reduced.

Two different types of error are included in the prediction: an offset error and a gain error. The offset error is introduced by the noninfinite gain of the comparator in Fig. 7 and the early effect, whereas the gain error is due to mismatch in the g_m of the current mirror transistors. Steps were taken to minimize offset errors and Monte Carlo simulations indicated that differences in gain would be the dominant source of prediction error.

This particular approach for prediction correction will correct for both types of error but only for one time interval at a time as it essentially applies an offset correction. Therefore if a new time delay is introduced the adaptive circuit must move to a new weight value. It is the long term goal of this work to implement the correction by adjusting the current sources in the current sink/source circuit (Fig. 7) so that the prediction can be correct over a range of time intervals. It is also worth noting that should a systematic error occur during spike generation at the pixels the prediction will adapt to accept that error. A random error will have no effect.

The STDP adaptation network has reduced the effect of process mismatch by "pulling together" spikes which are effectively coincident in real time, but have been spread on silicon by process imperfections.

VI. CONCLUSION

We have shown that STDP can be used to correct for process mismatch in an analogue VLSI system, using the signals that are naturally present in a spike-driven processing algorithm and without the need for an explicit calibration. We have demonstrated the success of the technique in the context of a neuromorphic, spike-time driven vision-processing algorithm, but the results have potential implications for all spike-timing processes and algorithms on silicon.

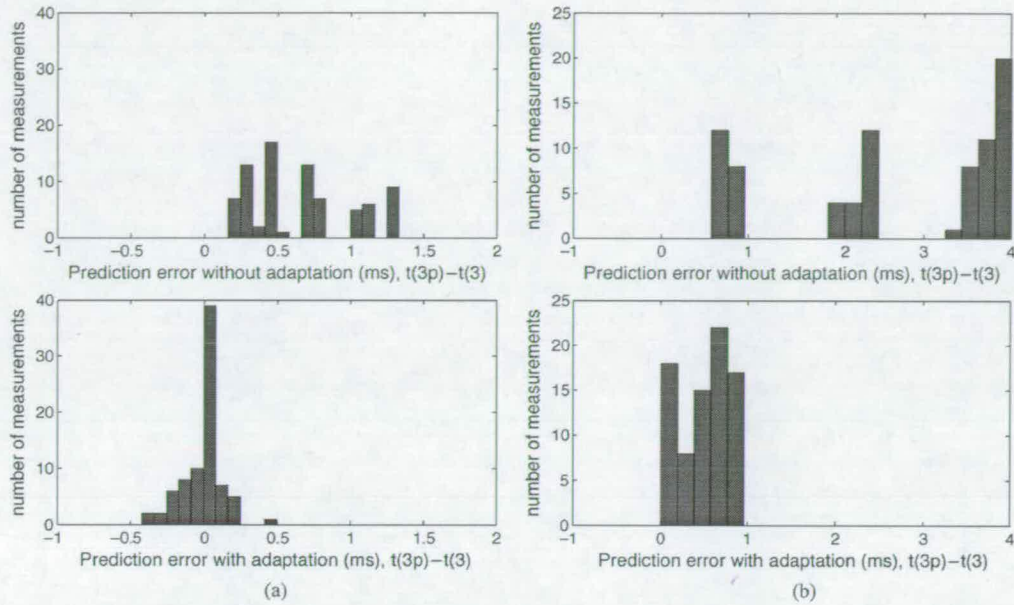


Fig. 16. Distribution of prediction errors from pixel activated circuit. The error measurements were taken with $V_{dd_mirr} = 3.3$ V. The current integrated on the LIF neurons came from the transient-detecting pixels. Results (a) were from a simulation when $t(2) - t(1) = 1$ ms and (b) $t(2) - t(1) = 5$ ms.

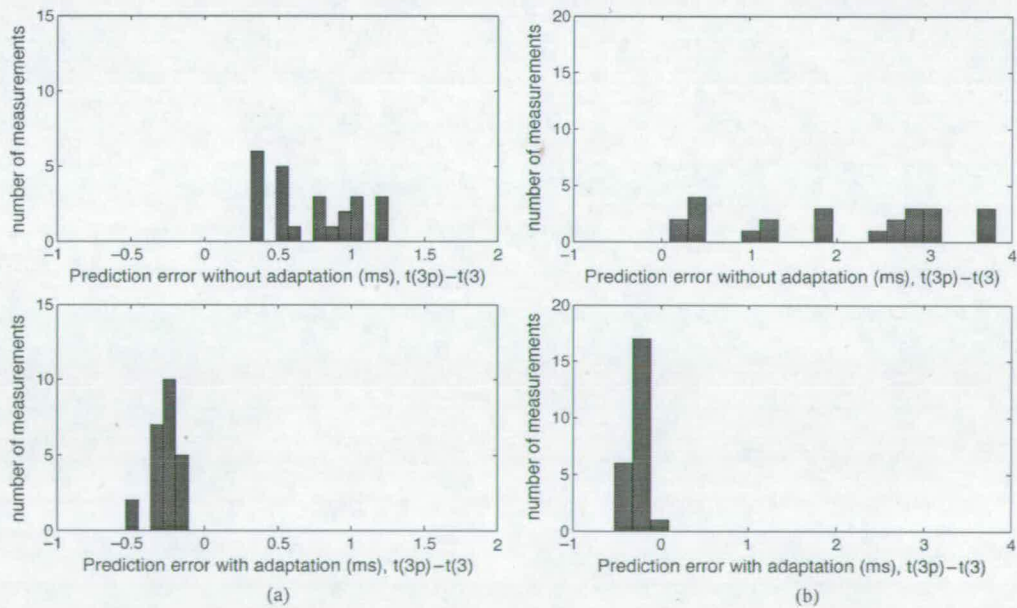


Fig. 17. Distribution of prediction errors from voltage controlled circuit. These results were recorded when $V_{dd_mirr} = 3.3$ V. Results (a) were from a simulation when $t(2) - t(1) = 1$ ms and (b) $t(2) - t(1) = 5$ ms.

TABLE I
MEAN AND STANDARD DEVIATION OF ERROR SPREAD STATISTICS

	Before Adaptation		After Adaptation	
	Mean (ms)	Sid (ms)	Mean (ms)	Sid (ms)
Pixel Driven 1ms Delay	0.65	0.36	-0.01	0.13
Pixel Driven 5ms Delay	2.62	1.27	0.52	0.27
Voltage Driven 1ms Delay	0.72	0.31	-0.27	0.08
Voltage Driven 5ms Delay	1.99	1.23	-0.26	0.11

While the STDP circuits are not small, they will shrink as process geometries shrink and the problem of mismatch becomes ever more bothersome. Furthermore, pixel-array applications are particularly area-sensitive and the technique may have immediate applications in tasks where the spike-processing circuits are themselves relatively large and the “overhead” of the STDP augmentation less of a concern.

This paper’s primary conclusion is, however, that STDP can be used to adapt system characteristics, in this case temporal characteristics “on the fly” to improve system performance. We have demonstrated that capability in 0.35 μm CMOS aVLSI.

APPENDIX PIXEL CIRCUIT ANALYSIS

A. Adapted Steady State

The transistor M_{fb} is operated in saturation and, for typical irradiance, in weak inversion. When the circuit is fully adapted to background illuminance ($V_{fb} \approx V_{diff}$) and neglecting early effects, we obtain

$$V_{fb} = \kappa_{fb}^{-1} \left(V_{ph} + V_T \log \left(\frac{I_{ph}}{I_{fb0}} \right) \right) \quad (1)$$

where I_{fb0} is the current-scaling parameter and κ_{fb} the subthreshold slope factor of M_{fb} . V_T denotes the thermal voltage kT/q , given by the absolute temperature T , the Boltzmann constant k and the elementary charge q . The voltage V_{ph} is set by the bias current through the inverting amplifier. Assuming a bias voltage V_b that puts M_n and M_p into weak inversion and again neglecting Early effects we obtain

$$V_{ph} = \kappa_n^{-1} \left(\kappa_p (V_{dd} - V_b) + V_T \log \left(\frac{I_{p0}}{I_{n0}} \right) \right) \quad (2)$$

where I_{n0} and I_{p0} are the current-scaling parameters of M_n and M_p , respectively, κ_n and κ_p are the corresponding subthreshold slope factors and V_{dd} is the potential of the positive power rail. In this approximation, V_{ph} is independent of I_{ph} . The actual dependence is due to the Early effects of M_n and M_p , i.e., to the limited gain of the inverting amplifier.

Neglecting the leakage currents of M_{ON} and M_{OFF} and assuming weak inversion and saturation for these transistors, we can compute the steady-state ON and OFF currents as

$$I_{ON} = I_{ON0} e^{\frac{(\kappa_{ON}(V_{amp} + V_{AB}) - V_{fb})}{V_T}} \quad (3)$$

$$I_{OFF} = I_{OFF0} e^{-\frac{\kappa_{OFF}(V_{amp} - V_{fb})}{V_T}} \quad (4)$$

where I_{ON0} and I_{OFF0} are the current-scaling parameters of M_{ON} and M_{OFF} , respectively, and κ_{ON} and κ_{OFF} are the corresponding subthreshold slope factors. The bias voltage V_{AB} for the class-AB differentiator can be expressed as

$$V_{AB} = \kappa_{AB}^{-1} V_T \log \left(\frac{I_{AB}}{I_{AB0}} \right) \quad (5)$$

where I_{AB0} is the current-scaling parameter and κ_{AB} the subthreshold slope factor of M_{AB} .

Leakage currents in the transient pathway affect, primarily, the diodes to the substrate, i.e., the source and drain diodes of M_{ON} and the well-to-substrate diode of M_{OFF} . The source diode of M_{OFF} does not contribute a leakage current because it is shorted and the leakage current of the drain diode of M_{OFF} is small. Furthermore, the parasitic currents at the drain diodes of M_{ON} and M_{OFF} do not influence the channel current, but add to the current drawn from subsequent devices. The source leakage current of M_{ON} and the well leakage current of M_{OFF} , however, result in a parasitic current I_{par} from the differentiator node into the substrate, which has to be balanced by an increased current through M_{ON} and, therefore, an increased gate voltage of M_{ON} with respect to V_{fb} . According to Kirchhoff’s current law, $I_{par} = I_{ON} - I_{OFF}$. Since I_{par} is typically large compared to I_{ON} and I_{OFF} , it dominates those currents ($I_{par} \approx I_{ON} \gg I_{OFF}$). Therefore

$$V_{amp} = \kappa_{ON}^{-1} \left(V_{fb} + V_T \log \left(\frac{I_{par}}{I_{ON0}} \right) \right) - \kappa_{AB}^{-1} V_T \log \left(\frac{I_{AB}}{I_{AB0}} \right). \quad (6)$$

If M_{ON} and M_{OFF} are implemented in the vicinity of the photodiode on the same silicon substrate, photo-induced minority carriers also contribute to the leakage and dominate it for large irradiance, such that it becomes roughly proportional to the photocurrent, i.e., $I_{par} = \delta I_{ph}$, where δ denotes the ratio of the electrons collected by the photodiode. The dependence of I_{par} on I_{ph} can be determined from the slope of the V_{amp} versus V_{fb} characteristic.

B. Transient

In the following, we will make a transient analysis of the circuit without considering parasitic capacitances and the effects of leakage currents in the transistors. The analysis presumes that the circuit variables have reached an equilibrium state before a transient change in the photocurrent is applied, and that no adaptation occurs in the considered time window.

The absolute value of the gain of the inverting amplifier is determined by the Early effects of M_n and M_p , and is given by

$$A_{amp} = -\frac{\partial V_{amp}}{\partial V_{ph}} = \frac{\kappa_n}{V_T} \left(V_{nE}^{-1} + V_{pE}^{-1} \right)^{-1} \quad (7)$$

where V_{nE} and V_{pE} are the Early voltages of M_n and M_p , respectively. Differentiating (1) and substituting $\dot{V}_{amp} = -A_{amp}\dot{V}_{ph}$ yields

$$\frac{\dot{I}_{ph}}{I_{ph}} = \kappa_{fb} \frac{\dot{V}_{fb}}{V_T} + A_{amp}^{-1} \frac{\dot{V}_{amp}}{V_T}. \quad (8)$$

Differentiating (3) and (4), respectively, gives

$$\frac{\dot{I}_{ON}}{I_{ON}} = \kappa_{ON} \frac{\dot{V}_{amp} + \dot{V}_{AB}}{V_T} - \frac{\dot{V}_{fb}}{V_T} \quad (9)$$

$$\frac{\dot{I}_{OFF}}{I_{OFF}} = -\kappa_{OFF} \frac{\dot{V}_{amp}}{V_T} + \kappa_{OFF} \frac{\dot{V}_{fb}}{V_T}. \quad (10)$$

If leakage currents in the differentiator stage are neglected, the capacitor current is given by

$$I_{ON} - I_{OFF} = C\dot{V}_{fb}. \quad (11)$$

It follows from (1) that:

$$\begin{aligned} I_{ON} - I_{OFF} &= \frac{C}{\kappa_{fb}} \left(\dot{V}_{ph} + V_T \frac{d}{dt} \log \left(\frac{I_{ph}(t)}{I_{fb0}} \right) \right) \\ &= \frac{C}{\kappa_{fb}} \left(\dot{V}_{ph} + V_T \frac{\dot{I}_{ph}}{I_{ph}} \right). \end{aligned} \quad (12)$$

In the closed-loop domain, where the feedback loop is activated, the \dot{V}_{ph} term can be neglected if the loop gain is much larger than unity ($\dot{V}_{fb} \gg \dot{V}_{ph}$). The difference in the transient currents is then proportional to the temporal derivative of the logarithm of the photocurrent, i.e., to the relative transient of the photocurrent.

ACKNOWLEDGMENT

The authors would like to thank P. Fleury for his assistance during chip layout and with tapeout. They would also like to thank A. Bofill, H. Chen, and S. Collins for their helpful discussions. The technical support of Austriamicrosystems and Europractice is gratefully acknowledged.

REFERENCES

- [1] M. J. M. Pelgrom, A. C. J. Duinaij, and A. P. G. Welbers, "Matching properties of MOS transistors," *IEEE J. Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1440, Oct. 1989.
- [2] F. Forti and M. E. Wright, "Measurement of MOS current in the weak inversion region," *IEEE J. Solid-State Circuits*, vol. 29, no. 2, pp. 138–142, Feb. 1994.
- [3] G. Wegmann and E. A. Vittoz, "Basic principles of accurate dynamic current mirrors," *Inst. Elect. Eng. Proc.*, vol. 137, no. 2, pp. 95–100, Apr. 1990.
- [4] J. Robert, P. Deval, and G. Wegmann, "Very accurate current divider," *Electron. Lett.*, vol. 25, no. 14, pp. 912–913, Jul. 6th, 1989.
- [5] Y. S. Yee, L. M. Terman, and L. G. Heller, "A 1 mV MOS comparator," *IEEE J. Solid-State Circuits*, vol. SC-13, no. 3, pp. 294–297, Jun. 1978.
- [6] W. Gerstner, "What's different with spiking neurons?," in *Plausible Neural Networks for Biological Modeling*. Norwell, MA: Kluwer, 2001, pp. 23–48.
- [7] W. Gerstner, R. Kempter, J. L. van Hemmen, and H. Wagner, "Hebbian learning of pulse timing in the barn owl auditory system," in *Pulsed Neural Networks*, W. Maass and C. M. Bishop, Eds. Cambridge, MA: MIT Press, 1998, ch. 14, pp. 353–377.
- [8] H. Hünig, H. Glünder, and G. Palm, "Synaptic delay learning in pulse-coupled neurons," *Neural Comput.*, vol. 10, pp. 555–565, 1998.
- [9] C. W. Eurich, K. Pawelzik, U. Ernst, J. D. Cowan, and J. G. Milton, "Dynamics of self-organized delay adaptation," *Phys. Rev. Lett.*, vol. 82, no. 7, pp. 1594–1597, Feb. 1999.
- [10] P. Häfziger, M. Mahowald, and L. Watts, "A spike based learning neuron in analog VLSI," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1997, vol. 9.
- [11] C. Gordon and P. Hasler, "Biological learning modeled in an adaptive floating-gate system," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 5, 2002, pp. 609–612.
- [12] A. Bofill, A. F. Murray, and D. P. Thomson, "Circuits for VLSI implementation of temporally-asymmetric Hebbian learning," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2002, vol. 14.
- [13] A. Bofill-i-Petit and A. F. Murray, "Learning temporal correlations in biologically-inspired aVLSI," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 5, May 25–28, 2003, pp. 817–820.
- [14] G. Indiveri, "Neuromorphic bistable VLSI synapse with spike-timing-dependent plasticity," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2003, vol. 15, pp. 1091–1098.
- [15] A. Bofill-i-Petit and A. F. Murray, "Synchrony detection by analogue VLSI neurons with bimodal STDP synapses," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2003, vol. 16.
- [16] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI reconfigurable network of integrate-and-fire neurons with spike-based learning synapses," in *Proc. Eur. Symp. Artificial Neural Networks*, Apr. 28–30, 2004, pp. 405–410.
- [17] A. Bofill-i-Petit and A. F. Murray, "Synchrony detection and amplification by silicon neurons with STDP synapses," *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1296–1304, Sep. 2004.
- [18] F. Wörgötter, A. Cozzi, and V. Gerdes, "A parallel noise-robust algorithm to recover depth information from radial flow fields," *Neural Comput.*, vol. 11, pp. 381–416, 1999.
- [19] T. Delbrück and C. A. Mead, "Adaptive photoreceptor with wide dynamic range," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 4, 1994, pp. 339–342.
- [20] J. Kramer, "An integrated optical transient sensor," *IEEE Trans. Circuits Syst. II, Analog Digit Signal Process.*, vol. 49, no. 9, pp. 612–628, Sep. 2002.
- [21] J. Goy, B. Courtois, J. M. Karam, and F. Pressecq, "Design of an APS CMOS image sensor for low light level applications using standard CMOS technology," *Analog Integr. Circ. Signal Process.*, vol. 29, pp. 95–104, 2001.
- [22] S. Loeda, "Circuits to support antisymmetric hebbian training in artificial neural networks," B.Eng. thesis, AU: WHAT DEPARTMENT?, Univ. Edinburgh, Edinburgh, U.K., May 2002.
- [23] K. Cameron and A. Murray, "Can spike timing dependent plasticity compensate for process mismatch in neuromorphic analogue VLSI?," in *Proc. IEEE Int. Symp. Circuits Systems*, vol. V, May 23–26, 2004, pp. 748–751.



Katherine Cameron (S'04) was born in Edinburgh, U.K., in 1979. She received the M.Eng. (Hons.) degree in electronics from the University of Edinburgh, Edinburgh, U.K., in 2002. She is currently working toward the Ph.D. degree in neuromorphic engineering at the University of Edinburgh.

Her current research interests include bio-inspired engineering solutions to analogue computation imperfections, mixed-signal VLSI design, and neural computation.



Vasin Boonsobhak (S'03) was born in Bangkok, Thailand, in 1975. He received the B.Eng. (Hons.) degree in electrical engineering from Mahanakorn University of Technology, Bangkok, Thailand, in 1996 and the M.Sc. degree in analogue and digital integrated circuit design from Imperial College, London, U.K., in 1998. He is currently working toward the Ph.D. degree in electrical engineering at the University of Edinburgh, Edinburgh, U.K.

Since 1996, he has been with the Department of Electronics Engineering, Mahanakorn University of Technology, Bangkok, Thailand. His current research interests include mixed-mode VLSI systems for image processing, image sensors and asynchronous digital interfaces for interchip connectivity.



Alan Murray (SM'94) was born in Edinburgh, U.K., in 1953. He received the B.Sc. degree (Hons.) in physics and the Ph.D. degree in solid state physics from the University of Edinburgh, Edinburgh, U.K., in 1975 and 1978, respectively.

He worked for three years as a Research Physicist and for three years as an Integrated Circuit Design Engineer. In 1984, he was appointed a Lecturer in electrical engineering at the University of Edinburgh. He became a Reader in 1991 and Professor of Neural Electronics in 1994. He is interested in all aspects of neural computation and hardware issues and applications have been his primary research interest since 1985. In 1986, he developed the "pulse stream" method for neural integration. His interests have since widened to include all aspects of neural computation, particularly hardware-compatible learning schemes, probabilistic neural computation and neural forms that utilize the temporal- and noisy characteristics of analogue VLSI—as well as applications of hardware neural networks. He is also developing a new interest in the interface between silicon and neurobiology, along with colleagues in Biomedical Sciences and in Glasgow University. He has over 200 publications, including an undergraduate textbook and research texts on neural VLSI, applications of neural networks and noise in neural training (with Peter Edwards).

Dr. Murray is a Fellow of the Institution of Electrical Engineers (IEE) and the Royal Society of Edinburgh, and a Member of INNS.



David Renshaw received the B.Sc. degree in pure mathematics, the M.Sc. degree in electronics, and the Ph.D. degree in very large scale integration (VLSI) design from the University of Edinburgh, U.K.

He is a Senior Lecturer in the School of Engineering and Electronics, University of Edinburgh, Edinburgh, U.K., where he has worked since 1981. His research interests include VLSI signal processing, analogue and digital CMOS circuit design, the design and manufacture of CMOS image sensors, machine vision and image processing applications.

In 1989, a small research team under his direction designed and demonstrated the first single chip CMOS video camera. In 1990, he was one of three founding members of VLSI Vision Ltd. (VVL) where he worked as Technical Manager, from 1990 to 1995. During that period, VVL grew from a research idea to 80 employees, became the leading supplier of CMOS imager chips and was placed on the London Stock Market. In 1996, he resumed full-time teaching and academic research in the areas of CMOS image sensors, image sensor-processors, image processing applications and CMOS mixed analogue-digital circuits, including, most recently, rapid prototyping of asynchronous circuit implementations of digital systems. He continues to support initial stages of company startup from these research activities.

A Neuromorphic Depth-From-Motion Vision Model With STDP Adaptation

Zhijun Yang, Alan Murray, *Senior Member, IEEE*, Florentin Wörgötter, Katherine Cameron, *Student Member, IEEE*, and Vasin Boonsobhak, *Student Member, IEEE*

Abstract—We propose a simplified depth-from-motion vision model based on leaky integrate-and-fire (LIF) neurons for edge detection and two-dimensional depth recovery. In the model, every LIF neuron is able to detect the irradiance edges passing through its receptive field in an optical flow field, and respond to the detection by firing a spike when the neuron's firing criterion is satisfied. If a neuron fires a spike, the time-of-travel of the spike-associated edge is transferred as the prediction information to the next synapse-linked neuron to determine its state. Correlations between input spikes and their timing thus encode depth in the visual field. The adaptation of synapses mediated by spike-timing-dependent plasticity is used to improve the algorithm's robustness against inaccuracy caused by spurious edge propagation. The algorithm is characterized on both artificial and real image sequences. The implementation of the algorithm in analog very large scale integrated (aVLSI) circuitry is also discussed.

Index Terms—Integrate-and-fire neurons, monocular depth recovery, neuromorphic vision model, synaptic plasticity.

I. INTRODUCTION

DISPARATE elements of the primate visual pathway, comprising, largely, the retina, thalamic relay neuronal network and visual cortex, cooperate to give rise to perceptive mapping of external stimuli. These perceptual properties have been attributed to complex brain activity at the cellular and network level [1]–[4]. In addition, experimental evidence on the activity-dependent firing behaviors of mammalian hippocampal pyramidal cells *in vitro* [5]–[7], tectal cells of a type of frog's visual system [8] and cortical neurons of rats [9] indicate that hippocampal, retinotectal and cortical neurons encode and process biological information by coordinating postsynaptic spikes with presynaptic inputs within a critical time window. The pre- and post-synaptic spike timing can thus decide the modification of input synaptic strengths, now called spike-timing-dependent plasticity (STDP). STDP suggests that long-term strengthening of synapses occurs if presynaptic action potentials precede postsynaptic firing by no more than about 50 ms, and long-term weakening of synapses occurs if

presynaptic action potentials follow postsynaptic spikes in the same time scale. This temporal correlation-based adaptation is speculated as a major learning rule in the nervous system (for reviews see [10], [11]).

Extensive theoretical and computational studies, based on the experimental results, have characterized the behavior of single neurons [12]–[15] or neuronal populations [16]–[20] in response to sensory stimuli. Many of these models are based on the widely used leaky integrate-and-fire (LIF) neuron model which, despite its simplicity, is capable of emulating some of the complex behavior of cortical spiking neurons [21]–[23]. Meanwhile, the effects of synaptic changes by STDP have become well established [24]–[26]. There are, however, few studies in which LIF networks with STDP adaptation mimic the biological perceptual functions for practical applications. Considering that vision processing is one of the most important functions of the primate brain, we explore in this paper one possible approach that mimics the early visual system.

Animal and human vision is able to detect moving irradiance edges and infer their associated depth in the image. It is well established that neurons in the primary visual cortex (area V1), a major part of the early visual cortex of primates, are able to detect edges in the visual field through their receptive fields [27], [28]. These neurons in V1 are retinotopically organized, following a roughly polar coordinate system [29]–[31]. Recent experiments also demonstrate that visual cortical neurons in V1 have the computational power to encode information about the three-dimensional (3-D) position of a stimulus in space [32], [33]. The outputs of V1 neurons contribute to the stereoscopic depth perception in the higher visual cortical areas [34], [35]. Although binocular stereopsis [36]–[38] is the most obvious way to recover depth, we can infer depth information using a single eye. In a dynamic environment, motion is a crucial cue in monocular depth recovery.

In applications of machine vision with motion, the scene is often modeled as an optical flow field, which may, in turn, be mapped onto a neuronal network [39]. One such simple network structure has neurons placed along axes arranged radially to the optical flow field center with only nearest-neighbor, on axis connections [40]. We aim to mimic the early visual processing in V1 with a simplified, continuous time vision model. The model uses a large-scale LIF neuronal network for edge detection and associated depth inference in a dynamic scene. The placement of LIF neurons in a polar coordinate arrangement bears some resemblance to the organization of V1 neurons. The characteristics of a dynamic scene are then reconstructed through local spike-timing computations along these radii.

Manuscript received July 13, 2004; revised August 18, 2005. This work is supported by EPSRC grant GR/R74567/01. This work is part of a joint EPSRC project involving the Universities of Edinburgh, Oxford, and Stirling.

Z. Yang is with the Department of Computer Science, Nanjing Normal University, Nanjing 210097, China (e-mail: zhijun.yang@ed.ac.uk).

A. Murray, K. Cameron, and V. Boonsobhak are with the School of Engineering and Electronics, The University of Edinburgh, Edinburgh EH9 3JL, U.K.

F. Wörgötter is with the Bernstein Centre of Computational Neuroscience, Göttingen 37099, Germany.

Digital Object Identifier 10.1109/TNN.2006.871711

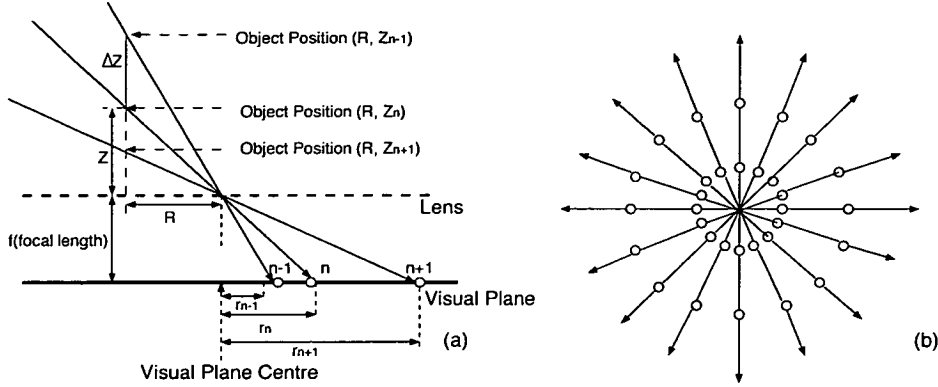


Fig. 1. Geometrical projection of a point object onto a visual plane of the proposed neuronal model via the light rays. (a) Side view, a 3-D point is described by the polar coordinate (R, Z) , R, Z are the distance from the point to the optical axis and lens, respectively. Z is the point depth which is usually lost in the 3-D to 2-D mapping. (b) Top view, a schematic layout of the visual plane, the small circles are neurons.

For computational simplicity, a neuron has only one synapse connecting with the photoreceptor adapted using an STDP rule. The advantage of employing STDP in this study is twofold. First, spike-timing related plasticity is a possible mechanism for pairing pre- and postsynaptic activities in a population of pulse-coded spiking neurons. In our algorithm, the postsynaptic firing depends on the timing of two excitatory inputs. Therefore, we will show that STDP is also able to optimize this form of vision model by adapting synapses online. As a result, neurons can process genuine edges while rejecting spurious edges that result from noise or the effects of an ill-conditioned optical flow field. Second, STDP can be implemented in analog very large scale integrated (aVLSI) circuits [41]–[43], and is capable of compensating for the inevitable transistor mismatch in aVLSI chip fabrication [44].

In summary, our neuron is fundamentally a simple detector of coincidence within a time window. The window size is determined by the neuron's dynamic threshold and input synaptic strength, which is adapted by the STDP rule. STDP adaptation improves the performance of this novel LIF-based network. It is also a neuromorphic system that mimics some aspects of the early visual pathway, especially the area V1 properties, in a new and interesting form. The experimental results will inform the development of aVLSI circuitry in subsequent works.

II. MONOCULAR DEPTH INFERENCE

Recently, Wörgötter *et al.* [40] devised a feature-based approach to monocular depth recovery from radial flow fields. The two key steps in their algorithm are: 1) calculation of the object coordinates in the environment and 2) a predictive mechanism to compare the predicted and actual time-of-arrival of an edge to improve noise rejection.

In their algorithm, the ego or object motion is restricted to a straight line along the optical axis with constant velocity. Neurons are aligned along the radial axes from the visual plane center to the periphery (see Fig. 1 for the geometric light ray projection and neuron layout). Upon the projection of a new edge

at a neuron, say $n-1$, neuron $n-1$ simply passes the gray level of that new edge to its succeeding neuron n , where it is saved. After some time, when the edge arrives at neuron n , the neuron compares its gray level with the saved one. If both levels match, neuron n treats the edge as having been successfully “confirmed once” and calculates the time-of-travel of the edge from neuron $n-1$ to n . From a known velocity, the distance that has been passed within the time-of-travel can be computed by neuron n , i.e., ΔZ in Fig. 1(a). Meanwhile, based on the temporal history of that edge, neuron n makes a prediction as to when neuron $n+1$ can expect the edge. This predicted time is transferred to neuron $n+1$ together with the gray level, and saved. On receiving the edge, neuron $n+1$ then compares it with the saved one. If both gray levels match, neuron $n+1$ continues to compare the predicted time and the actual time of its counter. If both predicted and actual time-of-travels match within a chosen tolerance, then the edge is treated as confirmed twice and is thus a more reliable edge. In the same way, neuron $n+1$ estimates the prediction time for neuron $n+2$ and enables the prediction-confirmation process to be repeated along the axis of neurons.

Clearly, a temporal tolerance or “window” is associated with the neurons. At an individual neuron, an edge is accepted if the predicted and actual time-of-travel of that edge are within the neuron's tolerance window. Otherwise the edge is rejected as spurious. Based on this monocular model, we present an early visual model incorporating STDP adaptation by using LIF neurons. We assume that the scenes will be mapped onto a visual plane to form an optical flow field through the retina. The optical flow field has edges flowing radially along axes of neurons, it comprises the spatio-temporal cues of objects in the three-dimensional space.

We consider a similar dynamic scene in which an observer moves along the optical axis toward the target objects at a constant velocity. The velocity of the projected edges on the visual plane increases with the distance from the visual plane center. The neurons in the visual plane are therefore arranged on the radial axes such that the distance between the consecutive pairs of neurons increases hyperbolically with increasing radius. With

TABLE I
SCENE AND RETINA DATA FOR EDGE DETECTION AND DEPTH INFERENCE

Scene Parameter	Value
Resolution	512x512 pixels
Gray level	8 bits
Edge contrast	0.10
Velocity	Constant value
Direction	Optical axis
Retina Parameter	Value
Visual plane α	4.025
Axes number	400
Neuron per axis	48 (artificial test)
Neuron per axis	74 (real test)
Retina radius	250 pixels

this arrangement, the time-of-travel of an edge between the consecutive pairs of neurons is approximately the same. Thus, the distance from each neuron to the visual plane center is

$$r_n = \sinh\left(\frac{n}{\alpha}\right) \quad (1)$$

where n is the neuron order which is a nature number increasing from the first neuron located near the visual plane center, and α determines the pixel resolution. After the placement of neurons in the visual plane, higher pixel resolution, if necessary, can be achieved by interpolating additional LIF units between existing neurons.

The depth information for a neuron is derived from simple geometric calculations [40]. In cylindrical polar coordinates we have

$$Z = \frac{\Delta Z}{k_n - 1} \quad (2)$$

where Z is the depth of a point object in the 3-D space, ΔZ is the distance moved from when the point activates neuron $n - 1$ until it reaches neuron n , see Fig. 1(a), $k_n = r_n/r_{n-1}$. The only part of the right-hand side of this depth equation that is not fixed for each neuron is ΔZ . Assuming that the ego or object motion velocity is constant and known, then ΔZ is also available by counting the time-of-travel of the projected edge between two successive neurons, and depth information can thus be recovered easily.

III. MODEL

We chose to use 400 identical radial axes to achieve a balance between image resolution and simulation time (and, ultimately, hardware cost). The LIF neurons have only nearest-neighbor, on axis connections. In the model we use a 512×512 pixel array which records the input image sequence at the same resolution. Forty-eight neurons are aligned on every axis for the simulation involving artificial data and 74 neurons per axis for the real image sequences. An edge is detected if a pixel has more than 10% irradiance difference compared to its surrounding pixels (see parameters in Table I).

A. Spiking Neuronal Model

Each neuron in this simplified model has three excitatory synapses that receive pulsed inputs. The synapses that carry a

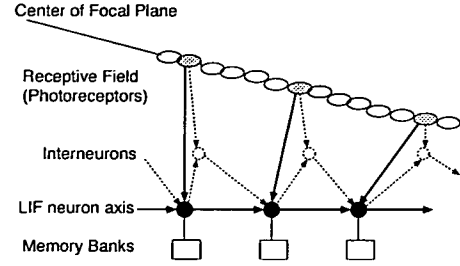


Fig. 2. Architecture of an axis of LIF neurons of the vision model. The photoreceptors align radially in the receptive field, they transfer the irradiance edge inputs to electrical pulse stimuli for the LIF neuron layer, the dotted photoreceptors have an actual connection with a corresponding neuron. A neuron is a basic computational unit with a memory. There is a layer of interneurons between the layers of photoreceptors and LIF neurons.

pulse from the preceding neuron and from the photoreceptor are referred to as the *flow synapse* and *receptive synapse*, respectively, see solid arrows in Fig. 2. The third input is used simply to start edge propagation via an interneuron, as described later. We make two assumptions for analytical and simulation simplicity: 1) there are no pulse transmission delays in the *flow synapses* between the LIF neurons and 2) the LIF neuron can not issue a spike unless its membrane is depolarized by both *flow* and *receptive synapses*.

The second criterion shows that an appropriate correlation between both *flow* and *receptive synaptic* inputs is a necessary condition for a LIF neuron to fire. A LIF neuron is a homogeneous leaky integrator. After detecting a new edge, its subthreshold membrane state in the presence of spikes from both *flow* and *receptive synapses* is

$$C_i^{\text{mem}} \frac{dv_i(t)}{dt} = g_i^{\text{mem}} (v_{\text{rest}} - v_i(t)) + \sum_{j=0}^1 I_j^{\text{syn}}(t) \quad (3)$$

where C_i^{mem} and g_i^{mem} are the membrane capacitance and conductance of neuron i , respectively. The last term of the right-hand side denotes summation of two excitatory postsynaptic currents (EPSCs). In the absence of the EPSCs and interneuron input, the membrane potential $v_i(t)$ will drift to its resting value v_{rest} . As the EPSCs drive the membrane potential above its threshold V_i^{th} , neuron i produces a short action potential and resets its potential to v_{reset} . We set $V_i^{\text{th}} > v_{\text{reset}} \geq v_{\text{rest}}$, and absolutely refractory period $t_{\text{ref}} = 0$. If neuron i detects a new edge, its interneuron is activated and emits a spike to the succeeding neuron $i + 1$. In that case, there is no EPSC from the *flow synapse* to neuron $i + 1$, which is, instead, depolarized by the interneuron.

The EPSC for both *flow* and *receptive synapses*, in our model, is a step followed by a rapid exponential decay [17]

$$\tau_{\text{syn}} \frac{dI_j^{\text{syn}}(t)}{dt} = -I_j^{\text{syn}}(t) + w_j S_j(t) \quad (4)$$

where τ_{syn} is the synaptic time constant, w_j is the strength of the *flow* or *receptive synapse* with $j = 0$ for *flow* and $j = 1$ for

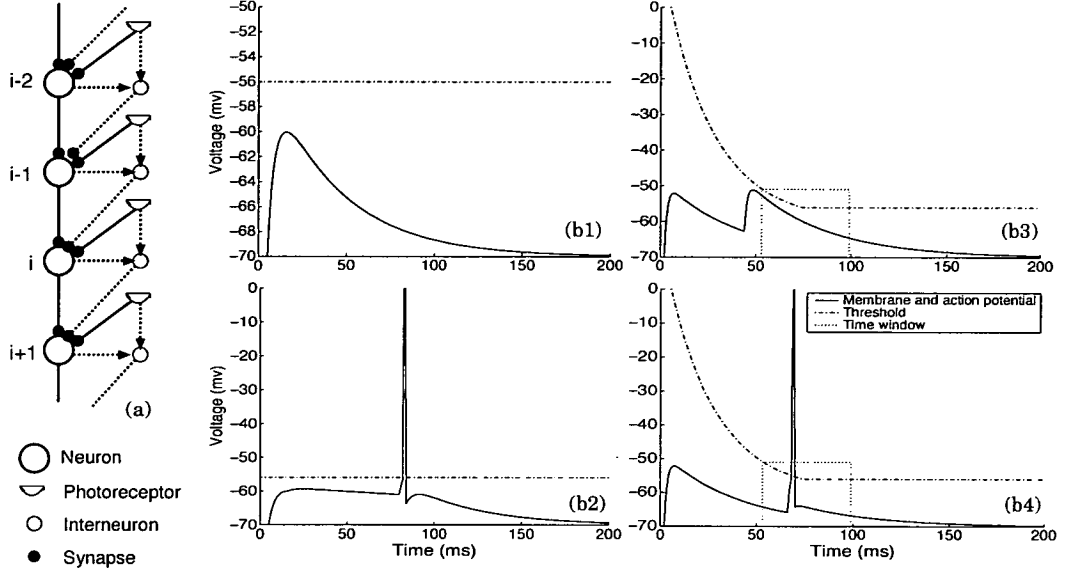


Fig. 3. Schematic diagram of an axis of spiking neurons and the possible membrane states. (a) Chain of neurons. Three synapses converge to a LIF neuron which has two outputs. (b1) EPSP on the membrane of neuron $i - 2$ caused by a spike from the *receptive synapse* representing the arrival of a new edge. (b2) Plateau membrane potential of neuron $i - 1$ caused by activation of the interneuron of neuron $i - 2$ and the subsequent firing due to the arrival of an edge at the photoreceptor; the spike encodes the time-of-travel of the edge starting from the onset of the plateau potential. (b3) Neuron i first receives a spike from its *flow synapse*, decoding the prediction of its firing in, say 76 ms; if an edge arrives at the photoreceptor but out of a tolerance time window of [53, 99], then neuron i does not fire. (b4) In a similar case, an edge arriving at 64 ms activates neuron i .

receptive synapse. $S_j(t)$ is a series of input spikes to synapse j that signal edges occurring at times t_j^k in the optical flow

$$S_j(t) = \sum_k \delta(t - t_j^k). \quad (5)$$

The concept of a dynamic threshold for pattern formation is biologically plausible [45], [46] and has already been used for synchronous spike generation in a neuronal network [18]. In our design, the threshold of a LIF neuron, i , is the greater of an exponentially decaying threshold $V_i^{\text{th}1}(t) = a_i \exp(-t/\tau_i^{\text{th}})$ and an intersecting constant threshold $V_i^{\text{th}2}(t) = V_T$

$$V_i^{\text{th}}(t) = \max \left(V_T, a_i \exp \left(-\frac{t}{\tau_i^{\text{th}}} \right) \right). \quad (6)$$

Here, a_i is the threshold value at $t = 0$, the time when the preceding neuron $i - 1$ fires, τ_i^{th} is the exponential threshold decay rate and V_T is the constant threshold value.

The intersection between $a_i \exp(-t/\tau_i^{\text{th}})$ and V_T , t_i^{pred} , is the time when neuron $i - 1$ predicts that an edge will reach neuron i . Since at that instant, $a_i \exp(-t/\tau_i^{\text{th}}) = V_T$, we thus have

$$t_i^{\text{pred}} = -\tau_i^{\text{th}} \ln \frac{V_T}{a_i} \quad (7)$$

and the threshold of a spiking neuron can be described in terms of t_i^{pred}

$$V_i^{\text{th}}(t) = \begin{cases} a_i \exp \left(-\frac{t}{\tau_i^{\text{th}}} \right), & \text{if } 0 \leq t \leq t_i^{\text{pred}} \\ V_T, & \text{if } t > t_i^{\text{pred}}. \end{cases} \quad (8)$$

An edge flowing along an axis of neurons is shown in Fig. 3. The interneurons are used to initialize the algorithm in the presence of new edges. If a new edge arrives at a photoreceptor connecting with neuron $i - 2$, then that neuron does not fire because it has not been depolarized by its *flow synapse*. In this case, the corresponding interneuron is activated and, hence, emits a spike to stimulate the succeeding neuron $i - 1$. Upon receiving a stimulus from the interneuron, the membrane of neuron $i - 1$ is depolarized to a plateau potential. This neuron will fire a spike whenever it receives a stimulus from its *receptive synapse*. If neuron i is depolarized by its *flow synapse* at time t , it is expected to be further depolarized by its *receptive synapse* within a time range of $(t + t_i^{\text{pred}} \pm \Delta t_i)$, where Δt_i is the tolerance window size of neuron i to be determined by the adaptation mechanism. If the depolarization from the *receptive synapse* is within the expected window, neuron i spikes and resets its membrane potential. The relative edge is then confirmed as a genuine feature by neuron i , and continues to flow along the axis.

t_i^{pred} is taken as the center of the tolerance window within which neuron i should receive an edge. We define the maximum and minimum tolerance window length as 60% and 10% of t_i^{pred} , respectively. Thus, we have $\Delta t_i^{\text{max}} = 0.3 t_i^{\text{pred}}$, $\Delta t_i^{\text{min}} =$

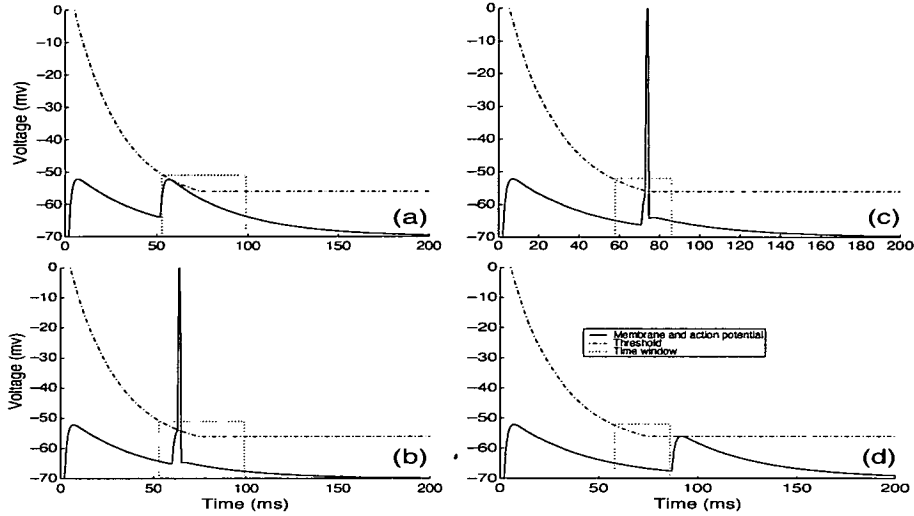


Fig. 4. Membrane dynamics of a neuron shows the temporal correlation of stimuli and response, and the effect of the STDP mechanism. The dynamic threshold consists of an exponential part and a constant part, they intersect at the prediction time when the neuron expects an edge from its *receptive synapse*. The first membrane depolarization is caused by the *flow synapse* spike, and the second by the edge from the *receptive synapse*. If the edge comes within a time window, then the neuron spikes. (a) Edge arrives at 52 ms. (b) Edge arrives at 60 ms. (c) Edge arrives at 71 ms. (d) Edge arrives at 87 ms. A time window of [53, 99] is applied for (a)–(b), and [58, 86] for (c)–(d). The prediction time for all figures is 75 ms. The test parameters are, the prediction parameter pair $\{\tau_{th}, w_{recept}\} = \{26 \text{ ms}, 13.91\}$ for (a) and (b), $\{26 \text{ ms}, 13.20\}$ for (c) and (d), the *flow synapse* $w_{flow} = 20$, the membrane time constant $\tau_{mem} = 40 \text{ ms}$, the synapse time constant $\tau_{syn} = 2 \text{ ms}$, the constant threshold $V_T = -56 \text{ mV}$, the neuron reset potential $v_{reset} = -65 \text{ mV}$, and the resting potential $v_{rest} = -70 \text{ mV}$.

$0.05t_i^{pred}$, the maximum possible firing window of neuron i is $(t + 0.7t_i^{pred}, t + 1.3t_i^{pred})$ and the minimum window is $(t + 0.95t_i^{pred}, t + 1.05t_i^{pred})$. For simulation simplicity, we only have the *receptive synapse* adaptable. An optimized window size for neuron i can then be determined by STDP adaptation of the *receptive synapse* (see Section III-C). If neuron i receives an edge from its photoreceptor before the predicted time t_i^{pred} , then its *receptive synapse* is increased and the window expanded. Otherwise its *receptive synapse* is depressed and the window is narrowed. The selection of appropriate exponential threshold decay time constants τ_i^{th} and *receptive synaptic* strengths are critical for the model. These two parameters are chosen and encoded by the preceding neuron $i - 1$ based on the prediction of time-of-travel of an edge. We simulate the spiking activities of a LIF neuron with a dynamic threshold first and then choose suitable values of τ_i^{th} and *receptive synaptic* strength to form the prediction parameter pairs, $\{\tau_i^{th}, w_{recept}\}$, for each possible predicted time in the prediction range. These parameters then determine the learning dynamics of the full vision model. Fig. 4 shows an example of the selected prediction parameter pairs, $\{\tau_i^{th}, w_{recept}\}$, with arrival of a sensory spike outside (a) and (d) or inside (b) and (c) a tolerance window whose size has been adjusted by changing the *receptive synaptic* strength alone.

B. Vision Algorithm Based on Spiking Neurons

Each model neuron unit has five major components; an edge-sensitive photoreceptor, a LIF neuron, *flow* and *receptive* synapses, and a dynamic short-term memory. The memory has the same basic function as the membrane capacitor of a

LIF neuron in terms of recording a spike event. Besides, we can use it to memorize some characteristics of a spike event such as an edge's gray level to facilitate software simulation. A LIF neuron also has a counter for the time course of the dynamic membrane threshold, and an additional input from an interneuron whose spiking status is determined by the states of the preceding LIF neuron and associated photoreceptor (Figs. 2 and 3). Image sequences with 512×512 pixel arrays (resolution) are used as the inputs.

Since the LIF neurons are *sparsely* allocated along radial axes, some photoreceptors have no connections with the LIF neurons, see the unfilled circles in the receptive field in Fig. 2. These photoreceptors are defined as the invalid photoreceptors. Accordingly, those photoreceptors connecting with LIF neurons are defined as the valid photoreceptors. In an initial period represented by a series of image frames starting from the first frame, there may be an edge, namely $e1$, projected onto the invalid photoreceptors which are located between two valid ones, $p1$ and $p2$. This edge has never passed $p1$, and it will pass $p2$. Before it arrives at $p2$, it is possible that the LIF neuron of $p2$ has had a plateau membrane potential which is induced by a new edge arriving at $p1$. In this case it is clear that, when edge $e1$ arrives at $p2$, it will, erroneously, force the LIF neuron of $p2$ to fire a spike based on the plateau potential which is not induced by $e1$ itself. We, hence, define the edges like $e1$ as the intermediate edges, and those edges which make the LIF neuron fire based on the plateau potential induced by themselves as the normal edges. After the algorithm starts, an initial self-organization stage can be seen. The convergence of depth inference occurs as soon as

all the intermediate edges become normal ones. We offer a proof of convergence of the algorithm in an Appendix.

When an edge, either normal or intermediate, arrives at the photoreceptor of neuron i , it induces an EPSP on the neuron's membrane. The sum of this EPSP, together with any EPSP induced by the *flow synapse*, determines the neuron's activity. If the total membrane potential is above the neuron's dynamic threshold, then the neuron emits a spike encoding the predicted time-of-travel after which that edge should arrive at the subsequent neuron, $i + 1$.

The predicted time-of-travel, t_{i+1}^{pred} , is defined by the temporal history of an edge arriving at neuron i , which can be described with the recursive equation

$$t_{i+1}^{\text{pred}} = \frac{t_i^{\text{actual}} + P t_i^{\text{pred}}}{P + 1} \quad (9)$$

where t_i^{pred} and t_i^{actual} are the predicted and actual time-of-travel of an edge arriving at neuron i from $i - 1$ respectively. P is the number of times that an edge has been correctly predicted as it propagates along the corresponding axis to neuron i . Upon receiving the prediction, neuron $i + 1$ then has $t_{i+1}^{\text{pred}} = -\tau_{i+1}^{\text{th}} \ln(V_T/a_{i+1})$ to determine the values of $-\tau_{i+1}^{\text{th}}$ and a_{i+1} for an appropriate dynamic threshold.

If no spike is emitted by neuron i upon arrival of an edge from its photoreceptor, then there are two possible reasons: 1) the edge is a new one, and the neuron's membrane has not yet been facilitated by a spike stimulus from its *flow synapse* and 2) the edge has arrived too early, or too late and is outside the tolerance window.

In either case, the relative interneuron emits a spike to neuron $i + 1$. The membrane of neuron $i + 1$ is thus facilitated, and the facilitation results in a plateau potential on the membrane of neuron $i + 1$ which will activate $i + 1$ on receiving an edge, intermediate or normal, from the photoreceptor at any time. The plateau potential is sustainable for a maximum duration of the time-of-travel of an edge from neuron i to $i + 1$.

C. Adaptation

We use STDP adaptation to improve this algorithm's robustness against inaccuracies caused by the collective behavior of edge flow, nonideal positioning of pixels as an array on the visual plane and inaccuracies from fabrication tolerances in an aVLSI implementation. The weight of the *receptive synapse* of each neuron is increased by a small amount if the *receptive synapse* fires before the predicted time instant and decreased if the *receptive synapse* fires late. The *flow synaptic* weight is held constant for simplicity. The adaptation scheme is

$$\Delta W_i(t) = \begin{cases} A_+ e^{-\frac{t - t_i^{\text{pred}}}{\tau_+}} & \text{if } t < t_i^{\text{pred}} \\ -A_- e^{-\frac{t - t_i^{\text{pred}}}{\tau_-}} & \text{if } t > t_i^{\text{pred}} \\ 0, & \text{if } t = t_i^{\text{pred}} \end{cases} \quad (10)$$

where $\Delta W_i(t)$ is the *receptive synaptic* weight change of neuron i , $\{A_+, \tau_+\}$ and $\{A_-, \tau_-\}$ are the amplitude and decay constant of potentiation and depression, respectively

TABLE II
MODEL NEURON DATA FOR EDGE DETECTION AND DEPTH INFERENCE

Neuron Parameter	Value
Membrane τ	40ms
Synapse τ	5ms
Adaptive τ_+	5ms
Adaptive τ_-	5ms
Rest potential	-70mv
Reset potential	-70mv
Constant threshold	-56mv
Flow weight	20
Adaptive amplitude A_+	0.2
Adaptive amplitude A_-	-0.3

and t_i^{pred} is the predicted time instant when neuron i should fire. The *receptive synaptic* weights are initialized to random values that correspond to a tolerance window width within the dynamic range, i.e., $(t + t_i^{\text{pred}} \pm \Delta t_i)$. If an edge arrives at the photoreceptor earlier than predicted, that edge has the potential to activate the neuron and the window is expanded to try to include such stimuli. If, however, an edge arrives later than predicted, it is likely to be uncorrelated with the predicted postsynaptic spike and the window is narrowed to exclude it. During the operation of the vision algorithm, the STDP process widens or narrows the tolerance window in order to include the correlated stimuli and exclude uncorrelated ones.

IV. SIMULATION RESULTS

This spike-based algorithm has been tested in both artificial and real environments. The neuron parameters are shown in Table II and we use 48 and 74 neurons per axis for the artificial and real images, respectively. All of the neuron parameters except the amplitudes of learning curves $A_+ = 0.2$ and $A_- = -0.3$ are chosen to be neurobiologically plausible. We choose A_+ and A_- to be larger than the values used in [24], as our pixel stimuli are considerably sparser. We use Euler integration as the numerical method [47] and the step is updated with each image frame.

A. Artificial Images

The artificial scene shown in Fig. 5 is constructed using OpenGL. Three objects, a cylinder, a cone and a sphere, are located at distances of 1500, 1100, and 700 cm, respectively, in front of a white background at 1600 cm. The observer is at a height of 1.6 m and moves toward the scene at a constant velocity of 1 cm per frame.

During motion, edges flow along the radial axes of neurons. Initially, in Fig. 5(a), the membrane of any LIF neuron whose photoreceptor detects a new edge is facilitated only by the *receptive synapse*. When a new edge is detected, the detecting neuron sends a spike representing the new edge to its neighboring neuron via the relative interneuron, which induces the plateau potential on the membrane of the neighboring neuron. The neighboring neuron then issues a spike when it receives a sensory edge at any time. The spike encodes and transmits the predicted time-of-travel of that sensory edge to the subsequent neuron on the axis. As the motion continues, any neuron activated by the correlated *receptive* and *flow synaptic* inputs issues

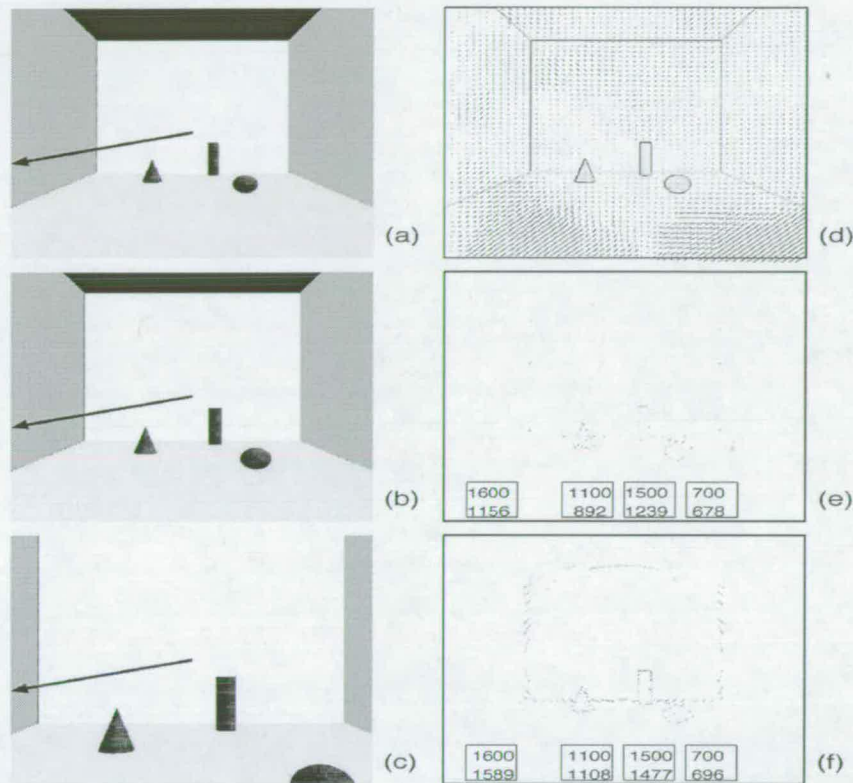


Fig. 5. Pixel maps of a hallway scene. (a)–(c) Snapshots of image frames at the start, 175 and 600 steps respectively, the arrow line is the approximate location of the 210th axis in each pixel map. (d) Still contour of the first frame without running the spike-based algorithm, the size of the short lines signifies the strength of optical flow in the 2-D space. (e)–(f) Pixel maps which are calculated by the algorithm from the start frame till the corresponding frame shown on the left. Each activated (black) pixel indicates the location of the first neuron which detected an edge and started the algorithm to encode the time-of-travel information for its subsequent neurons. If an edge flows reliably along an axis of neurons, that detecting neuron location can be recovered by a neuron far away from it. The depth boxes correspond to the depth inferred for the background frame, cone, cylinder and sphere, respectively, from left to right. Upper values in the boxes are the actual original depth of each object, lower values are estimated depth, both described in centimeters.

a spike, which represents the agreement of the actual and predicted time-of-travel, and hence, confirms the correct prediction of the time-of-travel of the relative edge. After the confirmation, the position of the detecting neuron is shown with black dot in Fig. 5(e) and (f), which form the reconstructed images corresponding to the scene before motion. We define these dots as the “effective pixels” and the images of these dots as the pixel maps.

Fig. 5(e) and (f) are examples of pixel maps for the artificial image test. Few edges appear in the early pixel maps, shown in Fig. 5(e), while the model is self-organizing (see appendix for details). After the initial settling period, all intermediate edges have become the normal edges in the sense that any predicted time-of-travel of an edge, given by the preceding neuron i , matches the actual time-of-travel of an edge arriving at neuron $i + 1$. It implies that an edge arriving at neuron i facilitates the membrane of neuron $i + 1$ and then, after a time-of-travel, the same edge will cause neuron $i + 1$ to fire. The detecting neuron of the confirmed edge is then included in the pixel map. The

counter of the firing neuron records the time-of-travel of the edge between the successive neurons. As the velocity is known, ΔZ in (2) is available and the depth of that edge can be recovered.

We can study the depth recovery process in more detail by concentrating on a single radial axis. In this simple example only one edge feature, the line where the left side wall and the back wall meet, will propagate along the chosen axis. The initial location of this edge feature is 16 m away from the camera. The depth information recovered by the neurons on the 210th axis is shown in Fig. 6(a) alongside the actual known depth of the edge in the image. The estimated depth matches the actual depth well, after an initial self-organizing process. The effect of the STDP mechanism can also be seen more explicitly through the distribution of the *receptive synapse* weights before and after adaptation. Initially weights are distributed randomly between maximum and minimum values [Fig. 6(b)]. After online STDP adaptation the initial uniform distribution converges toward the bimodal distribution of Fig. 6(c).

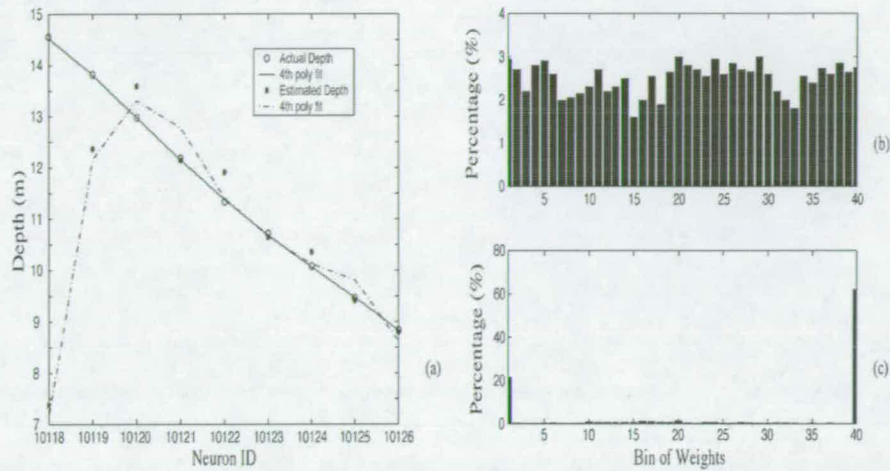


Fig. 6. Characteristics of the vision model operating on the hallway scene. (a) Comparison between the depth information estimated by the neurons on axis 210th and the corresponding actual depth, the data is fitted with a 4th degree polynomial function in a least-squares sense; the increasing order of neuron ID also represents the time course. (b) Initial random distribution of receptive synapse strengths in the weight range. (c) Bimodal distribution after STDP adaptation.

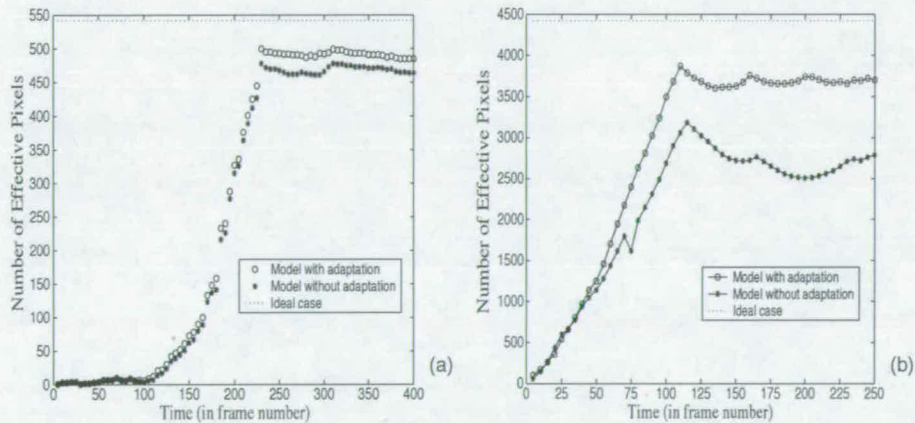


Fig. 7. Performance comparison of models with and without synaptic adaptation. The horizontal axis describes the time course with the number of image frames. The dotted lines in both figures are the ideal number of effective pixels. (a) Test on artificial images. The camera moves 1 cm/frame. Labels "o" and "*" represent the number of effective pixels detected by the models with and without adaptation, respectively, at a relative time instant. (b) Test on real images. The camera moves at 1.1 mm/frame with recording speed of 24 frames/s. Labels "o" and "*" represent the number of effective pixels detected by the models with and without adaptation, respectively.

The performance of the vision model can be shown quantitatively by comparing the number of effective pixels generated by the models with and without STDP adaptation. The comparison of performance of two models is shown in Fig. 7(a).

It is clear that, after the period of self-organization, both models reach a saturated stable state in which there are no significant changes in the number of edges detected by both models. The model with STDP adaptation is always superior. Both models take approximately 220 time units, equivalent to 2.2 m, to converge. After convergence, the model with adaptation has detected an average of 492 effective pixels, or 92.3%, out of a total sum of 533 effective pixels. The

model without adaptation has detected an average of 468 effective pixels, or 87.8% out of the same sum. The slight fluctuation observable after self-organization can be quantified as the mean and standard deviation of the number of effective pixels, (492, 4) for the model with adaptation, and (468, 5) for the model without adaptation. Although the performance of a model with adaptation is always better than that without adaptation, the difference is not worthy of note in the artificial test. This is intuitively correct, as the artificial image is, effectively, "perfect" and noise-free. The irregularities that our STDP adaptation process aims to reduce are, therefore, essentially absent.

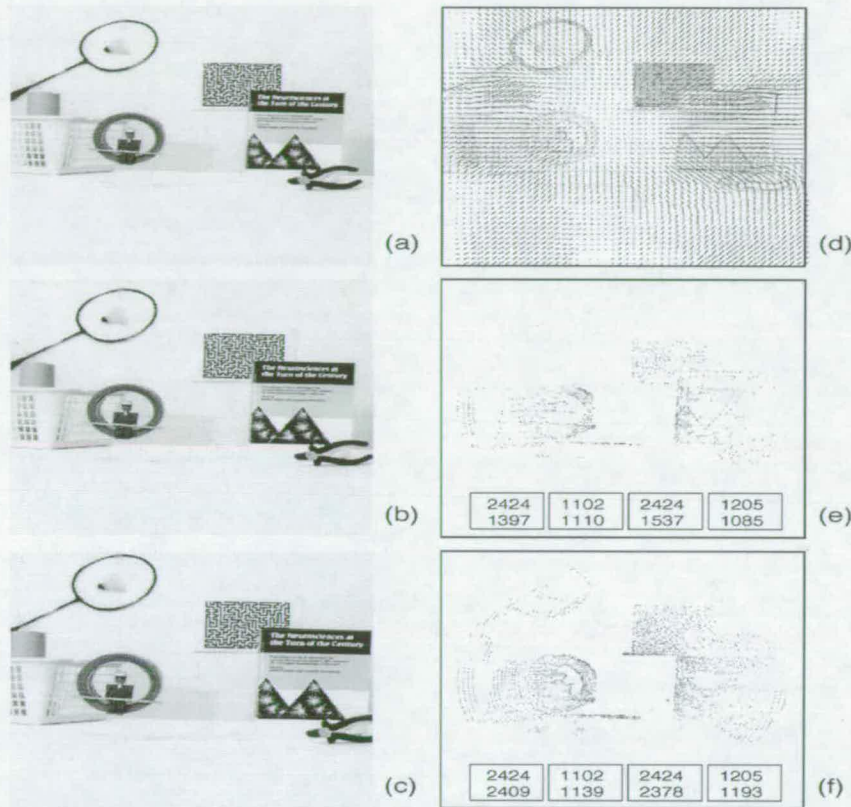


Fig. 8. Pixel maps of a real scene. (a)–(c) Snapshots of image frames at the start, 70 and 170 steps respectively. (d) Still contour of the first frame without running the spike-based algorithm, the size of the short lines signifies the strength of optical flow in the 2-D space. (e)–(f) Pixel maps. The depth boxes correspond to badminton racquet, toy pendulum, maze board and book, respectively, from left to right. Upper values in the boxes are the actual original depth of each object, lower values are estimated depth, both described in millimeters.

B. Real Images

In a real scene (Fig. 8), a book, a set of pliers, a maze board, a toy pendulum and a badminton racquet are arranged 1205, 1150, 2424, 1102, and 2424 mm away, respectively, from the initial camera plane. The lighting is primarily daylight, with two 150-W fluorescent lamps fixed on the ceiling and an additional 500-W studio light source to improve the recording contrast. A manually focused progressive scan NTSC CCD camera (model CV-M7 of JAI corporation, Japan) with Pentax Cosmicar 16-mm TV lens is mounted on a small board which is pulled by a small motor toward the objects. Before testing, the white balance of the camera is adjusted empirically. No other special procedure is adopted to adjust the camera optical and focus characteristics. The real scene is recorded in 250 frames with the camera motion speed approximately 1.1 mm/frame. The image frames are saved in a bitmap format with a resolution of 1276×1016 and 24 bits of color. The recording speed is 24 frames per second. The image frames are thereafter pre-processed to fit the final resolution of 512×512 and 8 bits of

gray level, and transferred to a SUN Blade 100 workstation for analysis.

The contours of the toy pendulum and book are easily recognizable from the pixel maps in Fig. 8. However, as the distance between edges in the maze approximates that of the pixel layout resolution, significant detail cannot be seen. The late appearance of the badminton racquet is caused by its location which is in the (peripheral) area of low neuron and pixel density. Depth information is recovered, as before, from the image sequence and the object depths are shown in Fig. 8. These results demonstrate that the algorithm works under real world conditions, once again assuming stable optical flow.

The ubiquitous noise and disturbance of optical flow can prevent the individual spiking neurons in a population corresponding to an object from having the equal inferred depth. A reliable method is to use the arithmetic average of the depth values computed by a group of neurons to depict an object's depth information. After an initial convergent period, the actual and estimated depth values agree quite well, see Fig. 8(f).

A performance comparison of the model working on the real image sequence, with and without STDP adaptation, is displayed in Fig. 7(b). After convergence, the model with adaptation has detected an average of 3689 effective pixels, or 83.4%, out of a total sum of 4421 effective pixels. The model without adaptation has detected an average of 2734 effective pixels, or 61.8% out of the same sum. Clearly the STDP adaptation plays an important role in improving the model performance.

The fluctuations in the number of effective pixels are a result of the inclusion and deletion method of the detecting neurons in the pixel map. A detecting neuron is included in the pixel map as long as the predicted time-of-travel of the relative edge is confirmed; it is excluded if any further confirmation fails in the subsequent neurons. The imperfect behavior that STDP ameliorates in this real image includes effects of camera tremble and disturbances of the scene context. We describe the fluctuation quantitatively with the mean and standard deviation of the number of effective pixels, (3689, 55) for the model with adaptation, and (2734, 181) for the model without adaptation. Our experiment shows that the model with adaptation is remarkably more robust in the real scene, where there are some imperfections to deal with.

The tests on both artificial and real images show that the distribution of all *receptive synaptic* strengths will display the bi-modal mode after adaptation; some of the *receptive synapses* tend to be depressed while the others potentiated. It implies that, given the radial neuron layout, the actual time-of-travel of edges between adjacent neurons may be consistently shorter (or longer) than the preceding neuron's prediction. This mismatch, namely layout mismatch, is caused by the inaccurate assignment of pixels (i.e., photoreceptor input) to the neurons during Cartesian to polar coordinate transform. This transform is necessary because the image sequences are in 512×512 Cartesian description while the visual plane uses polar coordinate to align neurons. Comparing the model performance with and without adaptation (Fig. 7), clearly STDP can compensate for this kind of mismatch remarkably well.

V. DISCUSSION

Using delays and coincidences, neural timing networks can perform time-domain signal processing operations to compare, extract and separate temporal patterns [48]. Temporal coding characteristics of spiking neuronal networks have been implemented in silicon circuits [49] to simulate some primitive functions of the early visual system, e.g., the orientation selective response of V1 cells [50], [51], and common recurrent networks [52]. Following this research trend, this study aims to provide a spike-timing dependent neuromorphic early visual model that emulates the function of the retina and area V1, which is thought to be capable of detecting edges and initiating depth inference. The model also aims to point toward a neuromorphic VLSI realization of the early visual processing in a dynamic scene. We therefore discuss other approaches to aVLSI motion detection along with the biological plausibility and neuromorphic feasibility of our model.

A. Some Two-Dimensional Neuromorphic Visual Motion Models

There are two major classes of motion detection algorithms, i.e., token-based and intensity-based approaches [53]. The correlation algorithm, a subdivision of the intensity-based class, measures spatio-temporal correlations caused by moving objects or ego-motion. Since correlation is based on multiplication, the correlation algorithms show superior numerical stability. However, most of these algorithms are unsuitable for implementation in integrated circuits that compute extended optical flow fields, because they are tuned to a narrow velocity range and tend to be expensive in silicon area [54].

For example, Delbrück's correlation-based, velocity-tuned retina was built on a hexagonal neuron grid and incorporated delay lines in three directions [55]. If the velocity of a moving edge matched a pre-set delay time of the delay line, the output signals grow in strength; otherwise they shrink. As the tuning is very narrow, the sensor is fairly limited, and considerable chip area is required to measure motion over a large velocity range. Similar work was also reported [56], but with the same limitations and is, hence, not suitable for the monolithic implementation of dense velocity-sensing arrays [54].

An arguably more practical system was proposed by Kramer *et al.* [57], [58] in the time domain. They presented two different circuits to measure the time-of-travel of an edge between two adjacent photoreceptors. One of which, i.e., the facilitate-and-sample (FS) motion method, is similar in principle to the decaying threshold mechanism in our vision algorithm. An equivalent approach was also implemented by Etienne-Cummings *et al.* [59], who suggested that this form of motion sensor may be more compact and thus suitable for use in a dense neuromorphic visual array.

Similarly, our algorithm measures the time-of-travel of an edge between two photoreceptors. This kind of method is more attractive for implementation in compact arrays [54]. Moreover, our model includes global edge detection at system level, as well as depth inference. Meanwhile, the study incorporates the real time synaptic strength adaptation mechanism to deal with the ubiquitous noise and mismatch problems, and the model performance is favorably displayed. None of the above methods address these issues.

B. Biological Plausibility

In different species of animals, different mechanisms reduce the optical flow to, if possible, a single component. Airborne animals such as birds and flies, tend to fly in straight lines. Under these conditions, only forward motion exists and optical flow is reduced to its radial components. Edges in the visual plane then move radially outwards from the focus of expansion with a velocity proportional to the radial distance from the focus of expansion but inversely proportional to their depth in the scene [40]. Hence, the depth can be inferred from this edge expansion which contains the temporal cues, i.e., the time-of-travel between adjacent photoreceptor locations in a retinotopically organized vision model.

Our model, which aims to simulate early visual mechanisms without the mediation of higher visual organizations, can be

seen to imply significant simplification of reality. Nevertheless, due to the adoption of an error-tolerance window mediated by STDP adaptation, we hope to relax the restriction of 100% constant velocity without compromising the model performance unduly.

C. Neuromorphic Feasibility

Presently the preliminary circuit simulation using Hspice in Cadence EDA tool has shown the possibility of neuromorphic realization of our model. We have included four new elements, plateau membrane potential, dynamic membrane threshold, counter and short-term memory, to the traditional LIF neuron. Three of them, except the memory, are essential for the hardware implementation of the algorithm. The memory is introduced to memorize the facilitation, either decaying EPSP or plateau potential, caused by an edge arriving at the preceding neuron. Considering that a silicon LIF neuron has a membrane capacitor which serves as a short-term memory to integrate input currents, we can simply ignore the memory design in the silicon neuron without undermining the vision model. The algorithm remains convergent, the proof in the Appendix still holds.

The concept of plateau potential is used to initiate the edge flowing along an axis of neurons. It is motivated by the experimental and analytical works revealing that a plateau potential can be triggered by depolarization in preceding neurons [60], [61]. In simulation, we use the output of an interneuron to generate the plateau potential on the membrane of a LIF neuron i . In hardware the effect of the interneuron can be replaced by an AND logic with two inputs from the output of neuron $i - 1$ and its associated photoreceptor, respectively. The output current of the AND logic charges the membrane capacitor of neuron i to an appropriate plateau voltage just below its (constant) threshold. Then neuron i can fire if it receives an edge from its *receptive synapse*.

When an edge is confirmed by a preceding neuron $i - 1$, the membrane potential of neuron i is facilitated by its *flow synapse*. The dynamic threshold is used to refrain neuron i from firing earlier than expected. For instance, starting from the firing time of neuron $i - 1$, neuron i should not be permitted to fire during an initial period to allow a time-of-travel of the relative edge. It is clear that the concept of dynamic threshold is closely related to the time-of-travel of an edge, which is measured by a counter. Suppose that the time-of-travel has been measured, in neuromorphic engineering the complementary metal-oxide semiconductor (CMOS) transistor circuits can be designed as a switch to restrict the firing window of a neuron. The window width can be adjusted by including the adaptation circuitry mediated by STDP.

The counter circuit of a neuron is the real concern in the neuromorphic implementation. As we introduced, many aVLSI circuits have been presented to measure the time-of-travels or velocities of edges between neurons during motion [55]–[59]. However, most of them are just the representation of the time durations or velocities in the sense that they have not used these values for further computing. In depth recovery we need the concrete time-of-travel information. Therefore, one option is that, for every neuron, we use an analog, calibrated current source

charging a capacitor as the time counter activated by a universal clock pulse generator, which generates a constant number of spikes per second, acting like a frame generator in a video camera device. The voltage of the capacitor can thus represent the time-of-travel of an edge, and can be used for further computing.

In summary, the proposed numerical model provides a possible approach to implement a monocular edge detector in aVLSI hardware, and certainly some modifications are necessary to implement the new elements which are not the traditional parts of a LIF model neuron. The output of the photoreceptors is binary, i.e., black and white, where black values for edges. The test results show that STDP adaptation can minimize the effects of process mismatch in the context of a spike-timing processing scheme for visual scenes. Further works will be focused on the two dimensional design of visual plane, including, e.g., communication between the photoreceptor array and visual plane processing circuit module and measurement of time-of-travel of edges.

VI. CONCLUSION

This study presents a neuromorphic approach to a large-scale, asynchronous, spiking network that will perform useful computation on a real visual scene. We have explored an artificial vision model based on LIF neurons for edge detection and depth analysis. The postsynaptic LIF neuron's activity depends on the correlation of two synaptic inputs. We demonstrate that the internal neuron parameters determine a temporal tolerance window within which the effect of correlations between the neuron's inputs affects its output. The window size is adjustable through synaptic strengths and membrane threshold time constants which are directly related to circuit parameters, rendering the model amenable to aVLSI application. As STDP is a likely adaptation mechanism underlying the visual cortical activities, we have introduced synaptic weight adaptation based on STDP, leading to a useful improvement in the model performance. Our approach to using STDP to ameliorate real system imperfections and deal with real data is general and is likely to have applications in other domains where timing is an intrinsic part of computation, such as audition.

APPENDIX

Hereby, we prove our belief that the optical flow of edges on the visual plane of spiking neurons will always converge, i.e., after a possible self-organizing period an edge passes spiking neuron i will be surely recognized by neuron $i + 1$ on the same axis in its traveling time of Δt_i between neurons i and $i + 1$. For simplicity, in the proof we assume that the traveling time between neuron pairs is the same, i.e., $\Delta t_i = \Delta t_{i+1} = \Delta t$. For cases in which this assumption does not hold, the proof is still valid.

Proposition 1: In the proposed spiking neuron-based vision algorithm, the optical flow of edges in the visual plane of spiking neurons will always converge.

Proof: Without loss of generality, we have two assumptions: 1) in an ideal, continuous and real scene each edge will pass, rather than skip over, a neuron before it can reach the next

neuron on an axis, and 2) no edges can escape from the axis or no new edges can add into the axis during the propagation of edges which initially exist on an axis.

The simplest case is that, at the starting time of the algorithm, there is no edge on the invalid photoreceptors. If the valid photoreceptor of neuron i detects an edge for the first time, neuron $i + 1$ has a plateau potential on its membrane and will be activated by any stimulus from its *receptive synapse*. It is clear that the edge coming from neuron i will be confirmed as a predicted edge at neuron $i + 1$ in a period of Δt_i . This temporal interval information will be encoded by the spike issued to neuron $i + 2$. Since all the consequent neurons on the same axis after neuron $i + 1$ will sum up the *receptive synapse* induced EPSP with the *flow synapse* induced EPSP on their membrane, it is obvious that the edge sourced from neuron i will be consistently confirmed as the predicted edge as it propagates further from neuron $i + 1$.

The more general case is that, after the algorithm starts, the valid photoreceptor of neuron i detects an edge for the first time whilst more edges are also detected by the invalid photoreceptors between neuron i and $i + 1$ where $i = 0, 1, \dots, k - 1$ and k is the total neuron number. In this case the key is to prove that a plateau potential event in a neuron's memory will finally be matched by its initiator edge rather than by other edges from the *receptive synapse*. In order to achieve that, we assume a series of neurons $\{n_0, n_1, n_2, \dots, n_{k-1}\}$ on an axis. $\{a, b\}_{n_i}$ represents an EPSP event on the membrane of neuron n_i . $a = -1$ for a plateau potential event, or a equals to any positive number for the time after which neuron n_i expects an EPSP from its *receptive synapse*. b is the time when the event itself is created on the membrane of neuron n_i . At the start of the algorithm, the membrane potentials of all neurons are at the resting value. If an edge arrives at the photoreceptor of n_i , we mark this time instant as Δt_i . As the algorithm runs, the EPSP events on the membrane of the sequent neurons will be $\{-1, \Delta t_i\}_{n_{i+1}}, \{\Delta t_{i+1} - \Delta t_i, \Delta t_{i+1}\}_{n_{i+2}}, \{-1, \Delta t_{i+2}\}_{n_{i+3}}, \{\Delta t_{i+3} - \Delta t_{i+2}, \Delta t_{i+3}\}_{n_{i+4}}, \{-1, \Delta t_{i+4}\}_{n_{i+5}}, \dots, \{(-1)^{k+1}(\Delta t_{i+k} - \Delta t_{i+k-1})^{1-(-1)^{k/2}}, \Delta t_{i+k}\}_{n_{i+k+1}}$, which satisfy

$$\begin{cases} \Delta t_i > \Delta t_{2n+1} > \Delta t_{2n} \\ \Delta t_{2n} > \Delta t_{2n-1} \end{cases} \quad (11)$$

where $n = k/2$ if k is even, and $n = k - 1/2$ if k is odd. Thus, we know that the time sequence, when the *receptive synapse* induced EPSPs occur, $\{\Delta t_i\}$, $i = 0, 2, \dots, k - 1$, is a monotonically increasing sequence with the upper limit of Δt . Hence, we have

$$\lim_{i \rightarrow \infty} \Delta t_i = \Delta t. \quad (12)$$

Since every new edge passing the photoreceptor of neuron i will generate a plateau potential event to neuron $i + 1$, the above equation means that gradually a plateau potential event on a neuron's membrane will finally be matched by its initiator edge, hence, the algorithm converges after a self-organizing period. ■

ACKNOWLEDGMENT

The authors would like to thank L. Smith and M. Dahlem for helpful discussions, N. Pugeault for real image sequence acquisition, and P. Hillman for help on preprocessing of the real image sequence

sequence

REFERENCES

- [1] M. A. Arbib and P. Erdi, "Précis of neural organization: structure, function, and dynamics," *Behav. Brain Sci.*, vol. 23, pp. 513–571, 2000.
- [2] U. S. Bhalla and R. Iyengar, "Emergent properties of networks of biological signaling pathways," *Science*, vol. 283, pp. 381–387, 1999.
- [3] M. Tsodyks and C. Gilbert, "Neural networks and perceptual learning," *Nature*, vol. 431, pp. 775–781, 2004.
- [4] K. Funke, Z. F. Kisvarday, M. Volgushev, and F. Wörgötter, "Integrating anatomy and physiology of the primary visual pathway: from lgn to cortex," in *Models of Neural Networks IV: Early Vision and Attention*, J. D. Cowan, E. Domany, and J. L. van Hemmen, Eds. New York: Springer-Verlag, 2002.
- [5] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, "Regulation of synaptic efficacy by coincidence of postsynaptic apss and epsps," *Science*, vol. 275, pp. 213–215, 1997.
- [6] G. Q. Bi and M. M. Poo, "Activity-induced synaptic modifications in hippocampal culture, dependence on spike timing, synaptic strength and cell type," *J. Neuroscience*, vol. 18, pp. 10464–10472, 1998.
- [7] D. Debanne, B. H. Gähwiler, and S. M. Thompson, "Long-term synaptic plasticity between pairs of individual ca3 pyramidal cells in rat hippocampal slice cultures," *J. Phys.*, vol. 507, pp. 237–247, 1998.
- [8] L. I. Zhang, H. W. Tao, C. E. Holt, W. A. Harris, and M. M. Poo, "A critical window for cooperation and competition among developing retinotectal synapses," *Nature*, vol. 395, pp. 37–44, 1998.
- [9] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, "Physiology and anatomy of synaptic connections between thick tufted pyramidal neurons in the developing rat neocortex," *J. Phys.*, vol. 500, pp. 409–440, 1997.
- [10] P. D. Roberts and C. C. Bell, "Spike timing dependent synaptic plasticity in biological systems," *Biological Cybernetics*, vol. 87, no. 5–6, pp. 392–403, 2002.
- [11] D. Yang and M. M. Poo, "Spike timing-dependent plasticity of neural circuits," *Neuron*, vol. 44, pp. 23–30, 2004.
- [12] M. V. Tsodyks and H. Markram, "The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability," *Proc. Nat. Acad. Sci.*, vol. 94, pp. 719–723, 1997.
- [13] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA: MIT Press, 2001.
- [14] C. Koch, *Biophysics of Computation—Information Processing in Single Neurons*. Oxford, U.K.: Oxford Univ. Press, 1999.
- [15] J. F. Feng and D. Brown, "Impact of correlated inputs on the output of the integrate-and-fire models," *Neural Comput.*, vol. 12, pp. 671–692, 2000.
- [16] M. Usher, H. G. Schuster, and E. Niebur, "Dynamics of populations of integrate-and-fire neurons, partial synchronization and memory," *Neural Comput.*, vol. 5, pp. 570–586, 1993.
- [17] J. J. Hopfield and A. V. M. Herz, "Rapid local synchronization of action potentials: toward computation with coupled integrate-and-fire networks," *Proc. Nat. Acad. Sci.*, vol. 92, pp. 6655–6662, 1995.
- [18] W. Gerstner, "Populations of spiking neurons," in *Pulsed Neural Networks*, W. Maass and C. M. Bishop, Eds. Cambridge, MA: MIT Press, 1999.
- [19] N. Brunel, "Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons," *J. Computat. Neurosci.*, vol. 8, pp. 183–208, 2000.
- [20] N. Levy, D. Horn, I. Meilijson, and E. Ruppin, "Distributed synchrony in a cell assembly of spiking neurons," *Neural Netw.*, vol. 14, pp. 815–824, 2001.
- [21] H. C. Tuckwell, *Introduction to Theoretical Neurobiology: Nonlinear and Stochastic Theories*. Cambridge, U.K.: Cambridge Univ. Press, 1988, vol. 2.
- [22] L. F. Abbott and T. B. Kepler, "Model neurons: from hodgkin-huxley to hopfield," in *Statistical Mechanics of Neural Networks*, L. Garrido, Ed. New York: Springer-Verlag, 1990.
- [23] T. W. Troyer and K. D. Miller, "Integrated-and-fire neurons matched to physiological f-i curves yield high input sensitivity and wide dynamic range," in *Computational Neuroscience: Trends in Research*, J. M. Bower, Ed. New York: Academic, 1997.

- [24] S. Song, K. D. Miller, and L. F. Abbott, "Competitive hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neurosci.*, vol. 3, no. 9, pp. 919–926, 2000.
- [25] M. V. Rossum, G. Q. Bi, and G. G. Turrigiano, "Stable hebbian learning from spike-timing-dependent plasticity," *J. Neurosci.*, vol. 20, no. 23, pp. 8812–8821, 2000.
- [26] R. Kempter, W. Gerstner, and J. L. van Hemmen, "Hebbian learning and spiking neurons," *Phys. Rev. E*, vol. 59, pp. 4498–4514, 1999.
- [27] D. Marr, *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information*. San Francisco, CA: Freeman, 1982.
- [28] V. Lamme, V. R. Rodriguez, and H. Spekreijse, "Separate processing dynamics for texture elements, boundaries and surfaces in primary visual cortex of the macaque monkey," *Cerebral Cortex*, vol. 9, no. 4, pp. 406–413, 1999.
- [29] G. Holmes, "The organization of the visual cortex in man," in *Proc. Roy. Soc. London Ser. B*, vol. 132, 1945, pp. 348–361.
- [30] G. Holmes, "The representation of the visual field in human striate cortex," *Arch. Ophthalmol.*, vol. 109, pp. 816–824, 1991.
- [31] S. A. Engel, G. H. Glover, and B. A. Wandell, "Retinotopic organization in human visual cortex and the spatial precision of functional MRI," *Cerebral Cortex*, vol. 7, pp. 181–192, 1997.
- [32] Y. Trotter and S. Celebrini, "Gaze direction controls response gain in primary visual-cortex neurons," *Nature*, vol. 398, pp. 239–242, 1999.
- [33] Y. Trotter, S. Celebrini, B. Striccanne, S. Thorpe, and M. Imbert, "Modulation of neural stereoscopic processing in primate area V1 by the viewing distance," *Science*, vol. 257, pp. 1279–1281, 1992.
- [34] B. T. Backus, D. J. Fleet, A. J. Parker, and D. J. Heeger, "Human cortical activity correlates with stereoscopic depth perception," *J. Neurophys.*, vol. 86, pp. 2054–2068, 2001.
- [35] B. G. Cumming and A. J. Parker, "Responses of primary visual cortical neurons to binocular disparity without depth perception," *Nature*, vol. 389, pp. 280–283, 1997.
- [36] N. Qian, "Binocular disparity and the perception of depth," *Neuron*, vol. 18, pp. 359–368, 1997.
- [37] C. Zhou and H. Tao, "Dynamic depth recovery from unsynchronized video streams," in *Proc. Int. Conf. Comput. Vis.*, vol. 2, 2003, pp. 351–358.
- [38] J. L. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vis.*, vol. 12, pp. 43–77, 1994.
- [39] H. A. Mallot, *Computational Vision—Information Processing in Perception and Visual Behavior*. Cambridge, MA: MIT Press, 2001.
- [40] F. Wörgötter, A. Cozzi, and V. Gerdes, "A parallel noise-robust algorithm to recover depth information from radial flow fields," *Neural Comput.*, vol. 11, pp. 381–416, 1999.
- [41] P. Häfner, M. Mahowald, and L. Watts, "A spike-based learning neuron in analog VLSI," in *Advances in Neural Information Processing Systems (NIPS)*. Cambridge, MA: MIT Press, 1996, vol. 9, pp. 692–698.
- [42] A. Bofill, A. F. Murray, and D. P. Thompson, "Circuits for VLSI implementation of temporally-asymmetric hebbian learning," in *Advances in Neural Information Processing Systems (NIPS)*. Cambridge, MA: MIT Press, 2002, vol. 9, pp. 1091–1098.
- [43] A. Bofill and A. F. Murray, "Synchrony detection and amplification by silicon neurons with stdp synapses," *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1296–1304, May 2004.
- [44] K. L. Cameron and A. F. Murray, "Can spike timing dependent plasticity compensate for process mismatch in neuromorphic analogue VLSI?," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 5, Vancouver, BC, Canada, 2004, pp. 748–751.
- [45] D. H. Perkel, "A computer program for simulating a network of interacting neurons: I. organization and physiological assumptions," *Comput. Biomed. Res.*, vol. 9, pp. 31–43, 1976.
- [46] R. Borisyuk, "Oscillatory activity in the neural networks of spiking elements," *BioSyst.*, vol. 67, pp. 3–16, 2002.
- [47] D. Hansel, G. Mato, C. Meunier, and L. Neltner, "On numerical simulations of integrate-and-fire neural networks," *Neural Comput.*, vol. 10, pp. 467–483, 1998.
- [48] P. A. Cariani, "Temporal codes and computations for sensory representation and scene analysis," *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1100–1111, May 2004.
- [49] S. C. Liu and R. Douglas, "Temporal coding in a silicon network of integrate-and-fire neurons," *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1305–1314, May 2004.
- [50] K. Shimonomura and T. Yagi, "A multichip avlsi system emulating orientation selectivity of primary visual cortical cells," *IEEE Trans. Neural Netw.*, vol. 16, no. 4, pp. 972–979, Apr. 2005.
- [51] S. C. Liu, J. Kramer, G. Indiveri, T. Delbrück, and R. Douglas, "Orientation-selective avlsi spiking neurons," in *Advances in Neural Information Processing Systems (NIPS)*. Cambridge, MA: MIT Press, 2002, vol. 2, pp. 1123–1130.
- [52] E. Chicca, D. Badoni, V. Dante, M. Andreagiovanni, G. Salina, L. Carota, S. Fusi, and P. Giudice, "A vlsi recurrent network of integrate-and-fire neurons connected by plastic synapses with long-term memory," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1297–1307, May 2003.
- [53] R. R. Harrison and C. Koch, "A robust analog vlsi reichardt motion sensor," *Analog Integr. Circuits Signal Process.*, vol. 24, no. 3, pp. 213–229, 2000.
- [54] R. Sarpeshkar, J. Kramer, G. Indiveri, and C. Koch, "Analog vlsi architectures for motion processing: from fundamental limits to system applications," *Proc. IEEE*, vol. 84, no. 7, pp. 969–987, Jul. 1996.
- [55] T. Delbrück, "Silicon retina with correlation-based, velocity-tuned pixels," *IEEE Trans. Neural Netw.*, vol. 4, no. 3, pp. 529–541, May 1993.
- [56] H. C. Jiang and C. Y. Wu, "A 2-D velocity- and direction-selective sensor with bjt-based silicon retina and temporal zero-crossing detector," *IEEE J. Solid-State Circuits*, vol. 34, no. 2, pp. 241–247, Feb. 1999.
- [57] J. Kramer, "Compact integrated motion sensor with three-pixel interaction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 4, pp. 455–459, Apr. 1996.
- [58] J. Kramer, R. Sarpeshkar, and C. Koch, "Pulse-based analog vlsi velocity sensors," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 44, no. 2, pp. 86–101, Feb. 1997.
- [59] R. Etienne-Cummings, J. Van der Spiegel, and P. Mueller, "A focal plane visual motion measurement sensor," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 44, no. 1, pp. 55–66, Jan. 1997.
- [60] R. E. Russo and J. Hounsgaard, "Short-term plasticity in turtle dorsal horn neurons mediated by I-type Ca^{2+} channels," *Neuroscience*, vol. 61, pp. 191–197, 1994.
- [61] V. Booth, J. Rinzel, and O. Kiehn, "A compartmental model of vertebrate motoneurons for Ca^{2+} -dependent spiking and plateau potentials under pharmacological treatment," *J. Neurophys.*, vol. 78, pp. 3371–3385, 1997.



Zhijun Yang was born in Yangzhou, China. He received the B.Eng. degree from Central China University of Science and Technology, Wuhan, China, in 1989 and the M.Sc. degree from Nanjing University, Nanjing, China, in 1992, both in electrical engineering. He received the D.Sc. degree from the Department of Computer Science, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, with a dissertation on the theory and application of machine learning, in 1999.

He worked as an Analog Electrical Engineer in Nanjing Panda Electronics Co. Ltd., Nanjing, China, for four years, where he was responsible for video signal circuit design. After receiving his D. Sc. Degree, he was a Senior Software Engineer for Motorola Global Software Center, responsible for developing an embedded Java project for mobile computing. In 2001, he worked as a Research Associate with the Department of Physics, Heriot-Watt University, Edinburgh, U.K. Since 2002, he has worked at Department of Computer Science, Nanjing Normal University, Nanjing, China. Since December 2002, he has also been with the Institute for Integrated Micro and nano Systems (IMNS) of Edinburgh University as a Research Fellow. His research interests include machine learning, mixed signal VLSI circuit design, and bio-inspired software development.



Alan Murray (M'91–SM'94) was born in Edinburgh, U.K., in 1953. He received the B.Sc. degree (hons.) in physics in 1975 and the Ph.D. degree in solid state physics in 1978, both from the University of Edinburgh.

He worked for three years as a Research Physicist (two in Canada), and for three years as an Integrated Circuit Design Engineer. In 1984, he was appointed a Lecturer in electrical engineering at Edinburgh University, became a Reader in 1991, and Professor of Neural Electronics in 1994. He is inter-

ested in all aspects of neural computation and hardware issues and applications have been his primary research interest since 1985. In 1986, he developed the "pulse stream" method for neural integration. His interests have since widened to include all aspects of neural computation, particularly hardware-compatible learning schemes, probabilistic neural computation and neural forms that utilize the temporal—and noisy characteristics of analogue VLSI—as well as applications of hardware neural networks. He is also developing a new interest in the interface between silicon and neurobiology, along with colleagues in Biomedical Sciences and in Glasgow University, Glasgow, U.K. He has over 200 publications, including an undergraduate textbook and research texts on neural VLSI, applications of neural networks and noise in neural training (with Peter Edwards).

Dr. Murray is a Fellow of the Institute of Electrical Engineers, U.K., a member of the International Neural Network Society (INNS), and a Fellow of the Royal Society of Edinburgh.



Katherine Cameron (S'04) was born in Edinburgh, U.K., in 1979. She received the M.Eng. (hons.) degree in Electronics from the University of Edinburgh, Edinburgh, U.K., in 2002, where she is currently working toward the Ph.D. degree in neuromorphic engineering.

Her current research interests include bio-inspired engineering solutions to analogue computation imperfections, mixed-signal VLSI design and neural computation.



Florentin Wörgötter is a Research Professor at the Bernstein Centre of Computational Neuroscience, Göttingen, Germany, and co-affiliated as a Professor at the Department of Psychology, University of Stirling, Stirling, U.K.

He has undertaken experimental neuroscience research since 1985 and theoretical neuroscience research since 1988. His main research goal is to bring perception and action together in artificial systems (computer vision systems, robots), building on the knowledge gained from theoretical neuroscience.

His research is focused on the visual system (perception) and on biophysical machine learning (action). He developed a neuromorphic VLSI algorithm for stereoscopic depth analysis, which is now (as an FPGA implementation) used in several industrial applications. This broad interdisciplinary research has lead to publications in *Nature*, *Trends in the Neurosciences*, *Journal of Neuroscience*, and others. In addition, he has published textbook chapters on computational neuroscience of the primary visual pathway. His past research interests were focused on early vision problems; current research is now centered on problems of data-fusion in the field of active vision as well as the integration of sensor and motor aspects.



Vasin Boonsobhak (S'03) was born in Bangkok, Thailand, in 1975. He received the B.Eng. (hons.) degree in electrical engineering from Mahanakorn University of Technology, Bangkok, Thailand, in 1996 and the M.Sc. degree in analog and digital integrated circuit design from Imperial College, London, U.K., in 1998. He is currently working toward the Ph.D. degree in electrical engineering at the University of Edinburgh, Edinburgh, U.K.

Since 1996, he has been with the Department of Electronics Engineering, Mahanakorn University of Technology. His current research interests include mixed-mode VLSI systems for image processing, image sensors and asynchronous digital interfaces for interchip connectivity.

AN ADAPTIVE VISUAL NEURONAL MODEL IMPLEMENTING COMPETITIVE, TEMPORALLY ASYMMETRIC HEBBIAN LEARNING

ZHIJUN YANG^{*,†}, KATHERINE L. CAMERON[†], ALAN F. MURRAY^{†,‡}
 and VASIN BOONSOBHAK[†]

^{*}Department of Computer Science, Nanjing Normal University,
 Nanjing 210097, China

[†]Institute for Integrated Micro and Nano Systems (IMNS),
 School of Engineering and Electronics, University of Edinburgh,
 Edinburgh EH9 3JL, UK

[‡]Alan.murray@ee.ed.ac.uk

Received 28 March 2005

Revised 5 March 2006

Accepted 9 March 2006

A novel depth-from-motion vision model based on leaky integrate-and-fire (I&F) neurons incorporates the implications of recent neurophysiological findings into an algorithm for object discovery and depth analysis. Pulse-coupled I&F neurons capture the edges in an optical flow field and the associated time of travel of those edges is encoded as the neuron parameters, mainly the time constant of the membrane potential and synaptic weight. Correlations between spikes and their timing thus code depth in the visual field. Neurons have multiple output synapses connecting to neighbouring neurons with an initial Gaussian weight distribution. A temporally asymmetric learning rule is used to adapt the synaptic weights online, during which competitive behaviour emerges between the different input synapses of a neuron. It is shown that the competition mechanism can further improve the model performance. After training, the weights of synapses sourced from a neuron do not display a Gaussian distribution, having adapted to encode features of the scenes to which they have been exposed.

Keywords: Spike timing dependent plasticity; early vision model; integrate and fire neuron; adaptation; image analysis; object depth recovery.

1. Introduction

Synchronised spiking neuron activity has been observed in the cerebral cortex and in the hippocampus, and postulated to be essential for animal and human cognitive behaviour.^{1,2} Experiments have demonstrated that blind patients can see a pattern of light, which corresponds to the pattern on an array of synchronised electrodes directly imposed on to their visual cortex.³ Analytical approaches^{4–6} and further experiments^{7,8} have therefore been undertaken to investigate whether neural populations can propagate time-locked or synchronous spikes for neocortical information processing. Some such studies link synchronisation to object representation in the visual cortex.⁸ This paper draws directly upon recent

experiments on activity-dependent firing behaviour in mammalian hippocampal pyramidal cells *in vitro*^{9–11} and tectal cells of *Xenopus* tadpoles¹² which indicates that neocortical neurons encode and process biological information by coordinating post-synaptic spikes with presynaptic inputs within a critical (millisecond-scale) time window. This temporal relationship between stimulus and response can be generated by adaptation of input synaptic weights, via a family of algorithms known as Temporally Asymmetric Hebbian Learning (TAH) or Spike-Timing-Dependent (synaptic) Plasticity (STDP).

Modelling studies, many of which employ integrate-and-fire (I&F) model neurons, have demonstrated the ability of neural networks to emulate

some neocortical phenomena. Furthermore, STDP adaptation in networks of neurons has been shown to improve temporal sequence learning, propagation and prediction.^{13–18} The synfire chains model, for example, consisting of groups of feedforward I&F neurons, is able to propagate synchronised spike volleys through neuron pools without loss of synchrony.^{5,19} The Suri model also shows that synchrony of spike propagation can be enhanced by STDP in a recurrent I&F network.¹⁷ These models are fundamentally related to the hierarchical cortex model originated by Hubel and Wiesel^{20,21} in the sense that their structure has an input layer which receives the input spike volleys and then transmits them into the processing layers.

Recent studies of the retinotopic mapping of the visual cortex have revealed that different areas of the visual cortex may be stimulated by different retinal sites,²² possibly due to the fact that receptive fields lie on a continuum across the surface of the visual cortex.²³ This hypothesis provides a novel way of understanding the relationship between the external stimulus of a moving scene and cortical responses without a hierarchical cortex structure. Rao and Sejnowski presented a biophysical model capable of dealing with moving input stimulus without hierarchical structures.¹⁶ However, the model is focused on simulating detailed neuronal dynamics rather than modelling visual cortical functionalities.

Based on Wörgötter *et al.*'s optical flow algorithm²⁴ and our previous study,^{25,26} we continue to explore event-driven neocortical circuitry for object discovery and associated depth estimation in a real scene. We have been shown that the temporal sequence of edges may be predicted and the prediction accuracy improved through STDP adaptation of interconnecting synapses. Given that in our previous model, a neuron has only one synapse connecting to its preceding neuron, it is natural to ask how the model will perform if a neuron has additional synaptic connections to different neurons on the preceding concentric layer. This paper investigates whether a vision model based on a more extensively-connected I&F neural net, with an additional competition mechanism between the input adaptive synapses, will result in a different (better or worse) performance. The motivation of such a model is twofold. Firstly, a multi-connected neural network is almost certainly more biologically plausible. Secondly, earlier

experiments showed that a neuron may predict an incoming edge wrongly while its neighbouring neurons on the same concentric layer are able to make correct predictions on a correlated edge. If they can be caused to “compete” to control the state of their successor neuron to cancel out the spurious prediction, then the model performance should be improved. It is this hypothesis that we set out to test.

The basic component of the model is a simplified I&F model neuron with multiple excitatory input synapses. Each I&F neuron has one synapse activated by an edge sensitive image sensor, to capture moving edges, representing intensity differences, in a dynamic scene. The remaining input synapses receive pulses from neighbouring neurons. Through adaptive synaptic weights and appropriate decay rate of the dynamic membrane threshold, a neuron is able to encode the edge propagation.^{25,26} A spiking “window” mechanism exists within which a neuron can fire a spike provided that its membrane capacitor is charged above its dynamic threshold. The window size is determined through competition of input synaptic weights which are adapted by STDP rule.

This paper is organised as follows. The optical flow algorithm, together with the details of the visual plane architecture, are described in Sec. 2. Section 3 provides numerical simulation results of the model performance on a real image sequence. This is followed by a brief discussion and conclusions regarding the model properties, simulation outcomes and their biological links in Sec. 4.

2. Methods

The I&F neurons are distributed on concentric layers of a visual plane, and along axes arranged radially from the optical flow field centre with connections to their nearest-neighbours on the preceding concentric layer (see Fig. 1). Each of the neurons receives input stimuli from its corresponding image sensor. When a point edge is moving with a constant speed toward the visual plane, it passes space locations I , $I + 1$, $I + 2$, and is thus projected onto neurons number i , $i + 1$ and $i + 2$ on the visual plane. The distances between the neighbouring neurons, i to $i + 1$, and $i + 1$ to $i + 2$ are arranged in the way that the time of travel of the projected edge from neuron i to $i + 1$ is equal to that from neuron $i + 1$ to $i + 2$.

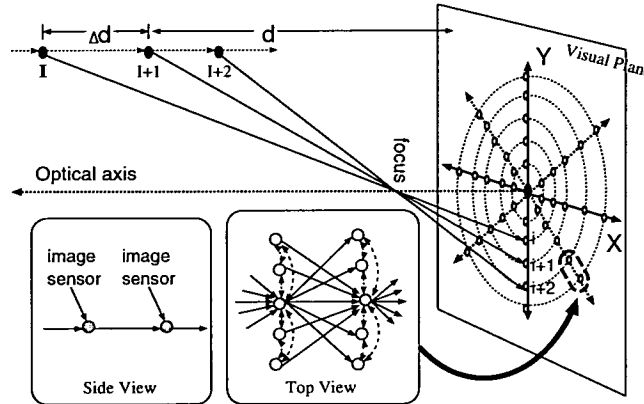


Fig. 1. The architecture of the visual plane model and schematic projection of objects onto the visual plane. A point object (solid circle) starts to move from location I to $I+1$ and then $I+2$, these 3 points are projected onto neurons i , $i+1$, $i+2$, respectively (open circles), on the visual plane. The neurons on the visual plane are arranged on concentric circular layers. Two insets give details to the connecting structure of two neighbouring neurons (dark circles) on one axis.

The derivation of the depth information for a neuron is a straightforward geometrical calculation.²⁴ In a 3-D space the depth of a point object to the visual plane is given by

$$d = \frac{\Delta d}{k_i - 1} \quad (1)$$

where Δd is the distance moved in the 3-D space from when the edge activates neuron i until the edge reaches neuron $i+1$, $k_i = \frac{r_{i+1}}{r_i}$, r_i is the distance from neuron i to the visual plane center. The only element on the right hand side of this polar coordinate equation that is not fixed for each neuron is Δd . Assuming that the ego/object velocity is constant and known, Δd is then available and depth information can be recovered readily. Once the depth has been recovered, a prediction of the time that the edge will reach $i+2$ can be calculated. The edge is only accepted as genuine if neuron $i+2$ fires within a given time window of this prediction. If neuron $i+2$ fires due to the arrival of an edge outside the time window, then that edge is rejected as a noisy signal.

2.1. Visual plane model

As it is shown in Fig. 1, every neuron in the visual plane model receives inputs from three different sources: its neighbouring neurons on the preceding concentric layer, the image sensor, and its neighbouring neurons on the same concentric layer.

The synapses connecting these different sources are referred to as *flow*, *receptive* and *lateral* synapses, respectively. Furthermore, the *flow synapses* are composed of *direct flow synapses* (i.e., on-axis connections) and *off-axis flow synapses* (see Fig. 2).

This visual plane model has a different architecture from that proposed previously^{25,26} in that neurons are now interconnected by two additional forms of synapses, i.e., the *lateral synapses* and *off-axis flow synapses*, whose functions will be described in detail later. In this new model, edges transmitted by the *off-axis flow synapses* are correlated with those transmitted by the corresponding *direct flow synapse* such that the *off-axis flow synapses* can widen the effective "capture area" of edges from the preceding concentric layer and improve rejection of spurious single-point effects. The *lateral synapses* are inhibitory which cancel out the excitatory effects of the relative *off-axis flow synapses* to allow the latter to contribute to adaptation for a better time prediction, without exciting the target neuron.

We assume a homogeneous visual model with notionally identical neurons which are leaky integrators whose subthreshold membrane state is

$$C_{mem} \frac{dv(t)}{dt} = g_{mem}(v_{rest} - v(t)) + \sum_i I_i^{flow}(t) - \sum_j I_j^{lateral}(t) + I^{recept}(t) \quad (2)$$

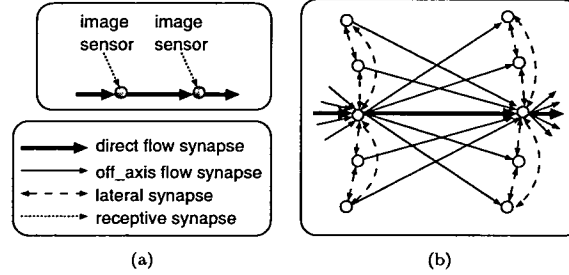


Fig. 2. A close view of four types of inter-neural connections in Fig. 1. (a) Side view of the connection structure of two neurons on an axis. (b) Top view of the connection structure of two neurons on an axis.

where C_{mem} and g_{mem} are the membrane capacitance and conductance of the neuron, respectively. The last three terms of right hand side denote the sum of input currents from the *flow*, *lateral* and *receptive synapses*, respectively. In the absence of all input currents, the membrane potential $v(t)$ will drift to its resting value v_{rest} . As the inputs drive the membrane potential above its threshold V_{th} , the neuron produces a short action potential and resets its potential to v_{reset} . In simulation we set $V_{th} > v_{reset} \geq v_{rest}$, and the refractory period $t_{ref} = 0$. When the synaptic time constants of all synapse types are equal, the dynamics of the three input current types are described by

$$\begin{cases} \tau_{syn} \frac{dI_i^{flow}(t)}{dt} = -I_i^{flow}(t) + w_i^{flow} S_i^{flow}(t) \\ \tau_{syn} \frac{dI_j^{lateral}(t)}{dt} = -I_j^{lateral}(t) - w_j^{lateral} S_j^{lateral}(t) \\ \tau_{syn} \frac{dI^{recept}(t)}{dt} = -I^{recept}(t) + w^{recept} S(t) \end{cases} \quad (3)$$

where $S(t)$ is the temporal sequence of spikes to the *receptive synapse*, whose weight is fixed in this study, $S_i^{flow}(t)$ and $S_i^{lateral}(t)$ are those to the *flow* and *lateral synapses* of neuron i . All spikes are modelled as events with a duration of zero. When a neuron fires, it issues a spike simultaneously to its on-axis neighbour and to the nearest off-axis neighbours on the next concentric layer, namely the on-axis neuron and off-axis neurons.

The weights of the *off-axis flow synapses* of neuron j are initialised to a Gaussian function of distance⁶ with regard to the weight of the *direct flow synapse* of neuron j . Thus the weight of a synapse

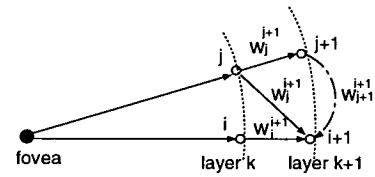


Fig. 3. Inter-neural connections in two neuron axes. Open circles represent neurons. Neuron $j+1$ is neuron j 's *on-axis* neuron, $i+1$ is j 's *off-axis* neuron, and vice versa. The synapses sourced from a neuron in the untrained visual net have a Gaussian weight distribution, $w_j^{i+1} > w_{j+1}^{i+1} \geq 0$. Lateral inhibition (dashed arrow line) with weight w_{j+1}^{i+1} acts to cancel out the activation effect of the *off-axis flow synapse* w_j^{i+1} to neuron $i+1$, i.e., $w_{j+1}^{i+1} = -w_j^{i+1}$.

sourced from neuron j to any interconnected neurons, say $i+1$, on the succeeding concentric layer has a general description, $w_j^{i+1} = \frac{w_j^{i+1}}{\sqrt{2\pi}\sigma} \exp \left(-\frac{|d_j^{i+1} - d_j^{j+1}|^2}{2\sigma^2} \right)$ (see Fig. 3). Here d_j^{i+1} and d_j^{j+1} are the Euclidean distances from a firing neuron to its on-axis neuron and to one of its off-axis neurons, respectively. If the Gaussian function has mean $\mu = 0$ and standard deviation $\sigma = \frac{1}{\sqrt{2\pi}}$, then we have

$$\begin{cases} w_j^{j+1} = w_j \\ w_j^{i+1} = w_j \exp \left(-\pi (d_j^{i+1} - d_j^{j+1})^2 \right) \end{cases} \quad (4)$$

From the above equation it is clear that the strengths of the *off-axis flow synapses* of a spiking neuron are defined with respect to its *direct flow synapse*. By using $w_j^{i+1} = -w_{j+1}^{i+1}$, the EPSP state of neuron $i+1$ is actually determined by the *direct flow synaptic weight* w_j^{i+1} . In simulation we set the

receptive synaptic weight as a constant dimensionless value of 20 for simplicity, and the initial direct flow synaptic weight $w_j = 15.8$. The model uses a biologically plausible, dynamic threshold for pattern formation.^{27,28} The threshold of an I&F neuron, i , is the greater of an exponentially decaying threshold $V_i^{th1}(t) = a_i \exp(-\frac{t}{\tau_i^{th}})$ and an intersecting constant threshold $V_i^{th2}(t) = V_T$, i.e.,

$$V_i^{th}(t) = \max\left(a_i \exp\left(-\frac{t}{\tau_i^{th}}\right), V_T\right) \quad (5)$$

where a_i is the membrane threshold value at $t = 0$, the time when the preceding neuron $i - 1$ fires, τ_i^{th} is the exponential membrane threshold decay rate and V_T is the constant membrane threshold value. The exponentially decaying membrane threshold represents a relative refractory period²⁹ to refrain neuron i from firing immediately after the activation of neuron $i - 1$. In our model design, the intersection between $a_i \exp(-\frac{t}{\tau_i^{th}})$ and V_T , t_i^{pred} , is the time when neuron $i - 1$ predicts, through its experience of edge arrival (see eqn.7), that an edge will reach neuron i . Since at that instant, $a_i \exp(-\frac{t}{\tau_i^{th}}) = V_T$. We thus have

$$t_i^{pred} = -\tau_i^{th} \ln \frac{V_T}{a_i} \quad (6)$$

If a neuron is de-polarised by its direct flow synapse at time t , we would expect it to be further de-polarised by its receptive synapse in a time window $(t + t_{pred} \pm \Delta t)$, where Δt is the window length to be determined by the adaptation mechanism. If the depolarisation from the receptive synapse is within the expected response window with respect to the spike from the direct flow synapse, then the neuron spikes and resets its membrane potential (see Ref. 26 also for details).

2.2. A vision prediction-confirmation scheme based on multi-connected spiking neurons

When an edge arrives at an image sensor connecting to the receptive synapse of neuron $i - 1$, it induces an EPSP on the neuron membrane. The sum of this EPSP, together with the EPSP induced by the direct flow synapse, determines the neuron's activity. If the total membrane potential is above the neuron's dynamic threshold, then the neuron issues a spike

to inform its succeeding neuron, i , that an edge will arrive.

Meanwhile, neuron $i - 1$ is able to estimate the predicted time of travel of the edge, t_i^{pred} , for neuron i . This temporal information is defined by the temporal history of the edge arriving at neuron $i - 1$, which can be described with the recursive equation

$$t_i^{pred} = \frac{t_{i-1}^{actual} + C t_{i-1}^{pred}}{C + 1} \quad (7)$$

where t_{i-1}^{pred} and t_{i-1}^{actual} are the predicted and actual time of an edge arriving at neuron $i - 1$ from $i - 2$ respectively. C is the number of "confirmation times" that an edge has been correctly predicted as it propagates along the corresponding axis to neuron $i - 1$.

Although in our model neuron the excitatory effect of off-axis flow synapses is cancelled out by the inhibitory lateral synapses, off-axis flow synapses still have influence on the postsynaptic neuron parameters. That means, the predicted time of travel of an edge, now T_i^{pred} , is decided by spikes from both direct flow synapse and off-axis flow synapses

$$T_i^{pred} = \frac{w_{i-1}^i}{w_{total}} t_i^{pred} + \sum_{offaxis} \frac{w_{offaxis}^i}{w_{total}} t_{i,offaxis,pred} \quad (8)$$

where

offaxis denotes the set of off-axis flow synapses which contribute to the modification of the predicted time of travel of an edge

w_{i-1}^i is the synaptic weight from neuron $i - 1$ to i

$w_{offaxis}^i$ is the synaptic weight from one of the off-axis neurons on the preceding concentric layer to neuron i

w_{total} is the sum of all flow synaptic weights which contribute to that neuron's spike generation

$t_{i,offaxis,pred}$ is the predicted time of travel of an edge sourced from one of the off-axis neurons on the preceding layer to i , which equals to the time from the off-axis neuron to its on-axis neuron.

The consequence of Eq. (8) is that a neuron's spiking activity depends upon all the states of the preceding neurons that are connected to it. It is then clear that Eq. (8) is a general case of Eq. (7) by incorporating off-axis synaptic inputs. Combining Eqs. (6)

and (8), it can be seen that all neurons that are “upstream” from a receiving neuron have an influence on its parameters

$$-\tau_i^{th} \ln \frac{V_T}{a_i} = \frac{w_{i-1}^i}{w_{total}} t_i^{pred} + \sum_{offaxis} \frac{w_{offaxis}^i}{w_{total}} t_i^{offaxis, pred} \quad (9)$$

If we substitute t_i^{pred} for the recursive Eq. (7), we can obtain a solution to the I&F neuron’s parameter pair, $\{\tau_i^{th}, w_{i-1}^i\}$, and therefore the visual plane model, including the dynamic threshold mechanism of its neurons, can be fulfilled.

2.3. Adaptation

In this study, STDP adaptation is used to render the model robust against inaccuracy, and hence to improve its depth-estimation performance. Inaccuracies may be caused by the collective behaviour of edge flow, by non-ideal positioning of image sensors as an array on the visual plane and by mismatches that are bound to result from fabrication tolerances in a silicon process. We use an STDP learning rule, that responds to spike synchrony within a time window.³⁰ The adaptation of each neuron potentiates or depresses its *flow synapses* by a small amount according to whether the depolarisation from a corresponding *flow synapse* is before or after the weighted prediction time instant. The *receptive synapse* weight is held constant. The adaptation of any individual flow synapse k converging to neuron i is

$$\Delta w_k = \begin{cases} A_+ e^{-\frac{t_k^{pred} - T_i^{pred}}{\tau_+}} & \text{if } t_k^{pred} < T_i^{pred} \\ -A_- e^{-\frac{T_i^{pred} - t_k^{pred}}{\tau_-}} & \text{if } t_k^{pred} > T_i^{pred} \\ 0 & \text{if } t_k^{pred} = T_i^{pred} \end{cases} \quad (10)$$

where Δw_k is the synaptic weight change of synapse k , which belongs to a set of synapses converging to neuron i , this set consists of one *direct flow synapse* from neuron $i - 1$ to neuron i , and all *off-axis flow synapses* converging to neuron i . $\{A_+, \tau_+\}$ and $\{A_-, \tau_-\}$ are the amplitude and decay constant of potentiation and depression, respectively. Initially, the untrained flow synaptic weights of each neuron are set to random values within the dynamic range of the neuron’s spiking window. During learning, the STDP process adapts the effective window through modifying the flow synaptic weights.

3. Simulation Results

The proposed model is tested using a real image sequence. In the experiment, object edges excite the I&F neurons, each of which transmits a spike through one *direct flow synapse* and eight *off-axis flow synapses*. The parameters chosen for this study are shown in Table 1. All of the neuron parameters except the initial amplitudes of learning curves $A_+ = 0.05$ and $A_- = -0.08$ are chosen to be similar to neurobiological exemplars. We choose A_+ and A_- to be larger than those used in an earlier study,³⁰ as our I&F neurons are considerably sparser. We use Euler integration as the numerical method³² and the time step is updated with each image frame.

In the scene, a book, a set of pliers, a maze board, a toy pendulum and a badminton racquet are arranged 1205, 1150, 2424, 1102, 2424 millimeters away, respectively, from the initial camera plane. The lighting is primarily daylight, with two 150 W fluorescent lamps fixed on the ceiling and an additional 500 W studio light source to improve the recording contrast. A manually focused progressive scan NTSC CCD camera (model CV-M7 of JAI corporation, Japan) with Pentax Cosmocar 16 mm TV lens

Table 1. Simulation parameters for object recognition and depth inference. The neuron parameters are adopted from Ref. 30.

Scene-retina parameter	Value	Neuron parameter	Value
Resolution	512 × 512 pixels	Membrane τ	20 ms
Focal parameter	4.025	Synapse τ	5 ms
Axes number	400	Rest potential	−70 mv
Neuron/Axis	74	Reset potential	−70 mv
Retina radius	250 pixel	Constant threshold	−56 mv

is mounted on a small board which is pulled by a small motor toward the objects. Before testing, the white balance of the camera is adjusted empirically. No other special procedure is adopted to adjust the camera's optics. The real scene is recorded in 250 frames with camera motion speed of approximately 1.1 mm per frame. The image frames are saved in bitmap format with a resolution of 1276×1016 and 24 bits of colour. The recording speed is 24 frames per second. The image frames are thereafter preprocessed to fit the final resolution of 512×512 and transferred to a SUN Blade 1500 workstation for analysis.

During camera movement, edges flow along the visual plane axes. Initially, in Fig. 4(a), all I&F activations are caused by the receptive synapses only and new events are transmitted, subsequently, to the neighbouring neurons via the *flow synapses*. As movement continues, any neuron activated by both the correlated *receptive* and *flow* synaptic inputs issues a spike. When a new edge is detected its presence at the current neuron is recorded. After this edge has been confirmed the position of the first detecting neuron is added to the pixel map as a black dot, see Figs. 4(e) & (f). The black dots are known as effective pixels.

The algorithm undergoes a "self-organisation" period as it converges to a stable stage, after which the number of the effective pixels remains approximately unchanged (this follows the method in Ref. 26). The stimuli from the *receptive synapses* begin to depolarise the membrane of the neurons after their image sensors detect optical edges. If the EPSPs induced by both the *receptive* and *flow* synapses of a neuron sum within the firing window to produce an activity that exceeds the combined threshold, then the neuron is activated, thus confirming the identification of the associated edge. The source neuron of the confirmed edge is then included in the pixel map and depth information can be calculated by the firing neuron according to Eq. (1). As further movement occurs in the scene and the confirmed edge continues to flow along the retinal axis it will be re-confirmed repeatedly as an ever more reliable representation of the edge.²⁴ In the pixel map of Fig. 4(f), the geometrical shape of objects is clear, and depth information can be recovered as described in Sec. 2. We use the arithmetic average of the depth values computed by a group of neurons to recover an

edge's depth information. This simple method can help to reduce the impact of inevitable noise.

Weight-independent STDP³⁰ is included to enhance the model's adaptation, as described in Sec. 2.3 above. When weight modifications are independent of the weight value, a bimodal weight distribution emerges from the learning process with weights reaching either the maximum or the minimum hard limits of synaptic strength due to the competition between synapses. This balanced form of bimodal weight distribution helps to stabilise the output rate.^{30,31} Figure 5(a) shows a comparison of performance between 3 different vision models, A, B and C in Table 2.

In Model A, the *off-axis flow synapses* compete with the *direct flow synapse* to control the post-synaptic neuron parameters. This model therefore takes better account of the collective behaviour of multiple inputs with similar spatiotemporal characteristics. The competition potentiates the more reliable synapses while depressing the less reliable ones according to Eq. (10). It is clear that, after an initial self-organisation period, the performance of the 3 models has become stable. With adaptation added in Models A and B, it was expected that a significantly improved performance would be measured. Model B confirms this by identifying approximately 35% more effective pixels than Model C, which is a basic model without adaptation and off-axis connections. Model A shows consistently superior performance to that of Model B by adding a further 7% improvement in the total number of the effective pixels. For the real image sequence, with the ideal case of 4421 edges projected on the neurons of the visual plane at the beginning, after convergence Model C can identify a total number of 2734 (62%) edges, Model B identifies 3689 (83%) edges, and Model A identifies 3875 (88%) edges.

In order to explain this further performance improvement, the weight distributions of the *direct* and *off-axis flow synapses* are shown in Fig. 5(b)–(d). Without loss of generality, all the weights of both the *direct* and *off-axis flow synapses* are initialised to their maximum possible values so that their expressions can be normalised to 1. The synapses of one axis of neurons, the 230th axis in this study, are used for analysis. After the initial period the weights of the *direct flow synapses* of some neurons are driven to the minimum hard limit while the others stick to the

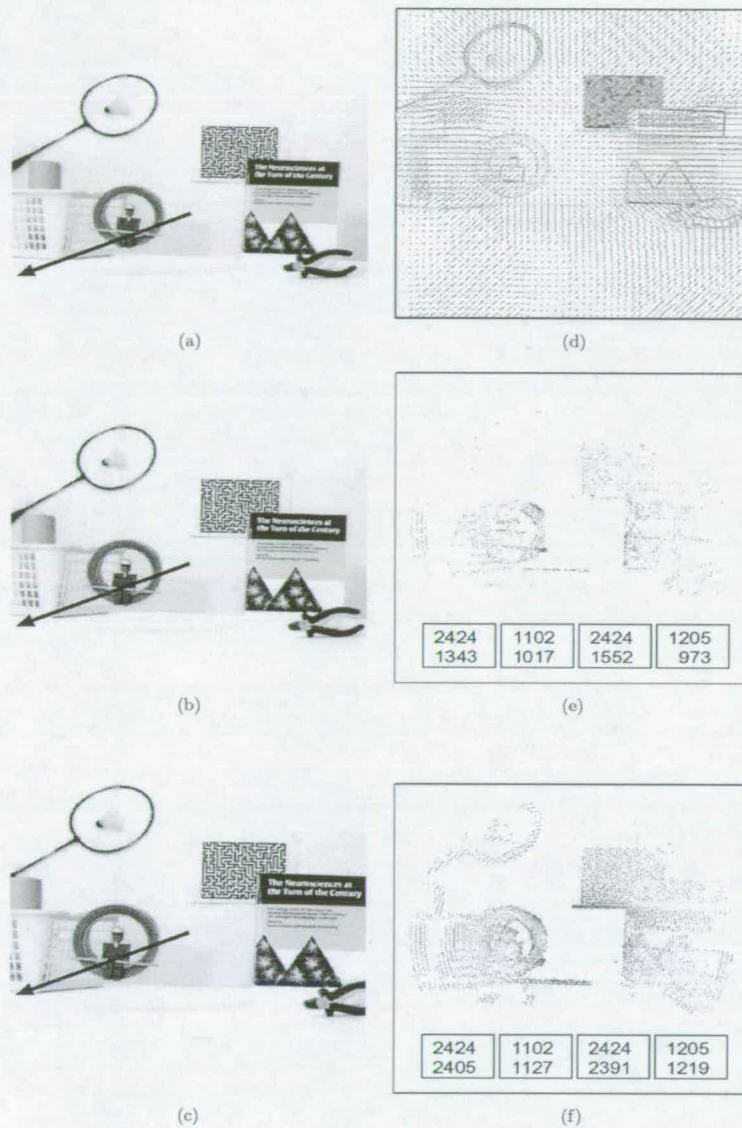


Fig. 4. The pixel maps of a real scene. (a)–(c) are the snapshots of image frames at the start, 60 and 200 steps respectively. The arrow represents the 230th axis which will be studied. (d) is the still contour of the first frame without running the spike-based algorithm, the short lines signifies the strength and direction of the optical flow in the 2D space. (e) and (f) are the pixel maps. The depth boxes correspond to the badminton racquet, toy pendulum, maze board and book, respectively, from left to right. Upper values in the boxes are the actual original depth of each object, lower values are the estimated depth, both described in millimeters.

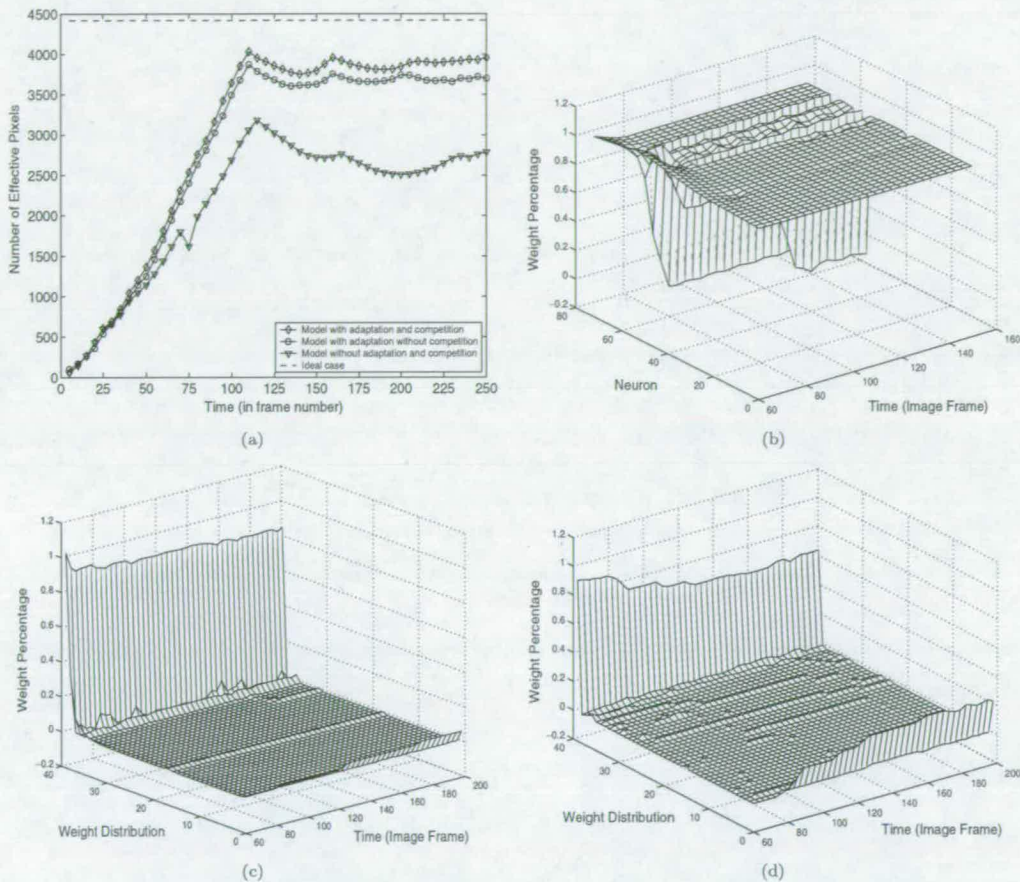


Fig. 5. Characteristics of the model performance and weight distribution. (a) A comparison of the performance of three different vision models. All the models experience a convergence stage and remain approximately stable despite the fluctuation. The dashed line represents the optimal result in which *all* the effective pixels, i.e., the edges projected onto the neurons of the visual plane at the start, are shown in an ideal pixel map. (b) The normalised weight values of the *direct flow synapses* of the 230th axis. They are initialised to their maximum values. As adaptation proceeds, some attain their minimum possible values and remain there. Most of the *direct flow synapses* of the neurons on the first half axis retain their maximum values because no stimulus is detected and no adaptation occurs. (c) The weight distribution of the *off-axis flow synapses* of the 230th axis. The emergence of weights distributed at the minimum hard limit is the result of competition between the *direct* and *off-axis flow synapses*. (d) The weight distribution of the *direct flow synapses* of the same axis. More of these synapses are driven to their minimum limit.

maximum hard limit, see Fig. 5(b). Figure 5(d) shows the weight evolution of the *direct flow synapses* of the 230th axis from the maximum to minimum hard limit in 40 bins. As the adaptation drives weights of maximum hard limit to the minimum limit, the weight distribution gradually forms a stable,

balanced bimodal format. The weight distribution of the *off-axis flow synapses* is shown in Fig. 5(c). Compared with Fig. 5(d), far fewer *off-axis flow synapses* are driven to their minimum hard limit after the initial period. This implies that a modest level of competition does occur between the *direct* and

Table 2. Characteristics of three different vision models.

	Model A	Model B	Model C
Off-axis synapses	✓	×	×
Adaptation	✓	✓	×
Input competition	✓	×	×

off-axis flow synapses. This allows us to explain the further performance improvement brought about by competition. Fundamentally, more competition leads to a more significant performance improvement because the competition mechanism strengthens synapses with the correct timing prediction, which then dominate the postsynaptic neuron parameters according to Eq. (9). The level of competition depends upon the image context and network structure. As an extreme example, in a scene with many feature curves whose curvature is the same as the curvature of the concentric neural layer, then multiple consecutive neurons on the same concentric layers can and will be activated simultaneously. They will then compete to determine the states of their succeeding neurons through their synapses. On the other hand, if a scene contains few feature curves that fall on or near the image sensor circles, the level of competition will be lower.

Normally, if an edge is correctly predicted and confirmed more than twice by the consecutive neurons on an axis, it can be treated as a genuine feature and hence its origin neuron location can be permanent in the pixel map. In this case the curves in Fig. 5(a) should be flat after the initial self-organisation period.

In order to test the influence of stochastic components in the optical flow field on performance, we have modified the edge inclusion criterion such that an edge is removed from a pixel map if it is judged by a neuron as a spurious edge, even if it has been confirmed by more than two neurons before. This criterion gives rise to the systematic fluctuation in the number of effective pixels in Fig. 5(a). The stochasticity can be caused by various reasons, such as tremble of the camera and disturbance of the scene context. However, our experiment shows that the stochastic fluctuation does not compromise the model's performance, and that, with the adaptation and competition mechanisms, the model is even more robust to such fluctuations.

4. Conclusions

We have introduced a new spike-based, adaptive visual plane algorithm that allows competition between on-axis and off-axis neurons to improve both the rate of acceptance of real edges and rejection of noise. We show that inter-synaptic competition results in a better model performance since the synapses carrying more "reliable" edge signals can win the competition to determine the activities of their postsynaptic neurons. It may therefore be desirable to have more synapses competing with their corresponding *direct flow synapse*. We are led to speculate that there are two factors which influence the level of competition in the model, i.e., the image context and neuron density on a visual plane. Clearly, the characteristics and curvature of the image can not be regarded as adjustable parameters. Neuron density seems, however, to be a major influence on the level of competition. If the neuron density is greater (i.e., there are many more axes), edges will inevitably create more off-axis signals and thus more competition.

When a neuron receives spike events from both the *direct* and *off-axis flow synapses*, weight-independent STDP included in the model introduces inter-synaptic competition to drive weights towards bimodal limits. Thus the initial Gaussian weight function of the synapses sourced from a neuron may not be sustainable.

The form of the weight distribution is also, clearly, affected by the characteristics of the features in the image. In the case of an "archery target" image of concentric circles, for example, competition will be maximised and the off-axis weight distribution tends to a balanced, bimodal form. In the case of an essentially random image, little competition will occur and the off-axis weights will remain at their initial (maximal) values.

In summary, we have presented a novel monocular vision model for object recognition and depth inference. As the underlying algorithm is based upon spike-timing and synchronisation, we have incorporated STDP adaptation to improve performance. In this study, adaptation includes lateral inter-neuron synapses, to allow information that is off the most obvious neuron axes to have a (restricted) influence on the fundamentally-axial computation. To our knowledge, this study is one of the very few works incorporating STDP adaptation, a new synap-

tic plasticity rule, into a vision model which is subsequently applied and measured in a vision case study. If the hard maximum and minimum limits are imposed on the synaptic weights, and weight modification is independent of its value, then STDP-induced competition will drive the weights to form a bimodal distribution. This weight redistribution helps representing the various information in a scene. We have demonstrated in our model that both the adaptation and competition can improve the model performance, although a law of "diminishing returns" inevitably applies as performance is driven towards 100%.

Acknowledgment

This work is supported by EPSRC grant GR/R74567/01. We thank Florentin Wörgötter, Leslie Smith and Markus Dahlem for helpful discussions.

References

1. P. N. Steinmetz, A. Roy, P. J. Fitzgerald, S. S. Hsiao, K. O. Johnson and E. Niebur, Attention modulates synchronized neuronal firing in primate somatosensory cortex, *Nature* **404** (2000) 187–189.
2. E. Salinas and R. Romo, Neurobiology: A chorus line, *Nature* **404** (2000) 131–133.
3. W. H. Doherty, M. G. Mladejowsky and J. P. Girvin, Artificial vision for the blind: Electrical stimulation of visual cortex offers hope for a functional prosthesis, *Science* **183** (1974) 440–444.
4. M. Abeles, *Corticonics* (Cambridge University Press, Cambridge, 1991).
5. M. Diesmann, M. O. Gewaltig and A. Aertsen, Stable propagation of synchronous spiking in cortical neural networks, *Nature* **402** (1999) 529–533.
6. C. Mehring, U. Hehl, M. Kubo, M. Diesmann and A. Aertsen, Activity dynamics and propagation of synchronous spiking in locally connected random networks, *Biol. Cybern.* **88** (2003) 395–408.
7. Y. Ikegaya, G. Aaron, R. Cossart, D. Aronov, I. Lampl, D. Ferster and R. Yuste, Synfire chains and cortical songs: Temporal modules of cortical activity, *Science* **304** (2004) 559–564.
8. R. Eckhorn, A. Gail, A. Bruns, A. Gabriel, B. Al-Shaikhli and M. Saam, Neural mechanisms of visual associative processing, *Acta Neurobiol. Exp.* **64** (2004) 239–252.
9. H. Markram, J. Lubke, M. Frotscher and B. Sakmann, Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs, *Science* **275** (1997) 213–215.
10. G. Q. Bi and M. M. Poo, Activity-induced synaptic modifications in hippocampal culture, dependence on spike timing, synaptic strength and cell type, *Journal of Neuroscience* **18** (1998) 10464–10472.
11. D. Debanne, B. H. Gähwiler and S. M. Thompson, Long-term synaptic plasticity between pairs of individual CA3 pyramidal cells in rat hippocampal slice cultures, *Journal of Physiology* **507** (1998) 237–247.
12. L. I. Zhang, H. W. Tao, C. E. Holt, W. A. Harris and M. M. Poo, A critical window for cooperation and competition among developing retinotectal synapses, *Nature* **395** (1998) 37–44.
13. L. F. Abbott and K. I. Blum, Functional significance of long-term potentiation for sequence learning and prediction, *Cereb. cortex* **6** (1996) 406–416.
14. W. Gerstner and L. F. Abbott, Learning navigational maps through potentiation and modulation of hippocampal place cells, *J. Comput. Neurosci.* **4** (1997) 79–94.
15. D. A. August and W. B. Levy, Temporal sequence compression by an integrate-and-fire model of hippocampal area CA3, *J. Comput. Neurosci.* **6** (1999) 71–90.
16. R. P. N. Rao and T. J. Sejnowski, Predictive sequence learning in recurrent neocortical circuits, *Advances in neural information processing systems* **12** (MIT press, 2000) 164–170.
17. R. E. Suri and T. J. Sejnowski, Spiking propagation synchronized by temporal asymmetric Hebbian learning, *Biol. Cybern.* **87** (2002) 440–445.
18. A. P. Shon, R. P. N. Rao and T. J. Sejnowski, Motion detection and prediction through spike-timing dependent plasticity, *Network: Comput. Neural Syst.* **15** (2004) 179–198.
19. M. Herrmann, J. A. Hertz and A. Prugel-Bennet, Analysis of synfire chains, *Network: Comput. Neural Syst.* **6** (1995) 403–414.
20. D. H. Hubel and T. N. Wiesel, Receptive fields and functional architecture of monkey striate cortex, *J. Physiol.* **195** (1968) 215–243.
21. D. H. Hubel and T. N. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, *J. Physiol.* **160** (1962) 106–154.
22. J. S. Sunness, T. Liu and S. Yantis, Retinotopic mapping of the visual cortex using functional magnetic resonance imaging in a patient with central scotomas from atrophic macular degeneration, *Ophthalmology* **111** (2004) 1595–1598.
23. D. L. Ringach, Mapping receptive fields in primary visual cortex, *J. Physiol.* **558** (2004) 717–728.
24. F. Wörgötter, A. Cozzi and V. Gerdes, A parallel noise-robust algorithm to recover depth information from radial flow fields, *Neural Computation* **11** (1999) 381–416.
25. Z. J. Yang and A. F. Murray, A biologically plausible neuromorphic system for object recognition and depth analysis, in *Proc. Europ. Symp. Art. Neural Networks* (Brugge, Belgium, 2004), pp. 157–162.

26. Z. J. Yang, A. F. Murray, F. Wörgötter, K. L. Cameron and V. Boonsobhak, A neuromorphic depth-from-motion vision model with STDP adaptation, *IEEE Trans. Neural Networks* **17**(2) (March, 2006), in press.
27. D. H. Perkel, A computer program for simulating a network of interacting neurons. I. Organization and physiological assumptions, *Comput. Biomed. Res.* **9** (1976) 31–43.
28. R. Borisyuk, Oscillatory activity in the neural networks of spiking elements, *BioSystems* **67** (2002) 3–16.
29. R. Sarpeshkar, J. Kramer, G. Indiveri and C. Koch, Analog VLSI architectures for motion processing: From fundamental limits to system applications, *Proceedings IEEE* **84** (1996) 969–987.
30. S. Song, K. D. Miller and L. F. Abbott, Competitive Hebbian learning through spike-timing-dependent synaptic plasticity, *Nature Neuroscience* **3**(9) (2000) 919–926.
31. R. Kempter, W. Gerstner and J. L. van Hemmen, Intrinsic stabilization of output rates by spike-based Hebbian learning, *Neural Computation* **13** (2001) 2709–2741.
32. D. Hansel, G. Mato, C. Meunier and L. Neltner, On numerical simulations of integrate-and-fire neural networks, *Neural Computation* **10** (1998) 467–483.

CAN SPIKE TIMING DEPENDENT PLASTICITY COMPENSATE FOR PROCESS MISMATCH IN NEUROMORPHIC ANALOGUE VLSI?

Katherine Cameron and Alan Murray

School of Engineering and Electronics
The University of Edinburgh
Edinburgh, EH9 3JL, UK

k.l.cameron@sms.ed.ac.uk, alan.murray@ee.ed.ac.uk

ABSTRACT

Analogue VLSI can be used to implement spike timing dependent neuromorphic training algorithms. This paper presents novel circuitry that uses spike timing to “adapt out” the effects of device mismatch in such circuits. Simulation results for the circuit implemented in a 0.35 μm CMOS process are reported.

1. INTRODUCTION

As circuit geometries decrease, the problems caused by device mismatch increase [1]. When sub-micron devices are operated in sub-threshold mode the effect is exacerbated [2]. The effect of threshold (V_{th}) variation can clearly be minimised by the use of transistor dimensions greater than minimum size. However, other compensation mechanisms have been proposed, which allow the fabrication of devices such as current mirrors and dividers and comparators [3]–[5] with improved matching characteristics.

We present a new technique for device mismatch correction based upon a form of asymmetric Hebbian training, a neural algorithm implementing Spike Timing Dependent Plasticity (STDP). In STDP synaptic weight change is driven not by spike-rate correlations, but by *individual inter-spike timings*. This class of algorithms has been shown to recover different and potentially richer information from a spike train when compared to rate and population coding methods [6] and due to its relative simplicity has been successfully implemented in analogue VLSI [7]–[11].

We have chosen, as a context for this study, a spike-timing driven algorithm for visual scene analysis [12] which, if implemented in analogue focal-plane VLSI, will be degraded by unavoidable mismatches in the temporal characteristics of nominally identical circuits. The algorithm is thus an excellent demonstrator for any benefits that STDP adaptation may offer. The algorithm is based upon the comparison of two related inter-neural firing times, where the delay between them effectively measures the speed of motion of an object in the visual field. A simple analogue circuit has been designed that predicts a spike arrival time from the timings of two previous spikes. This circuit has been subjected to “process mismatch” simulation and an adaptation network has been added to attempt to compensate for the mismatch error.

This paper will present the circuit designed and Monte Carlo simulation results showing both the problems caused by mismatch and the benefits of the adaptation network. A 0.35 μm CMOS chip is being fabricated to verify the simulation results.

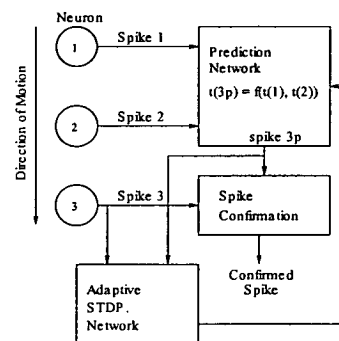


Fig. 1. Neuron firing time and prediction. Neurons 1 to 3 fire at predetermined intervals. Neuron 3p fires at a time predicted from the firing time of neurons 1 and 2. If the neurons fire at equal intervals $t(3p) = t(2) + \{t(2) - t(1)\}$ where $t(1)$ is the firing time of spike 1, $t(2)$ for spike 2 etc. Spike 3p is the prediction of spike 3.

2. PREDICTION CIRCUITRY

The system chosen to use spike timing dependent training to adapt out device mismatch is that of Wörgötter et al [12] in which the time at which an edge passes a pixel in a visual radial flow field is used to predict further such events. This can be used to eliminate noise by accepting only events that have been predicted using signals from previous edge-sensitive pixels. In this paper, neuron firing times will replace the edge times. The details of the vision-processing algorithm are not relevant as the presented circuit implements the noise reduction part of the work and not the actual depth-from-motion algorithm.

The system block diagram is shown in figure 1. The spike firing circuit is made up of three “integrate and fire” neurons that are set to fire at predetermined intervals. For example, the delay between neurons one and two firing may be the same as the delay between neurons two and three. In a complete system the intervals would be determined by pixel spacing. The spikes from neuron 1 and 2 are passed to the prediction network which generates another spike, spike 3p (i.e. “predicted”, not actual), at a time determined

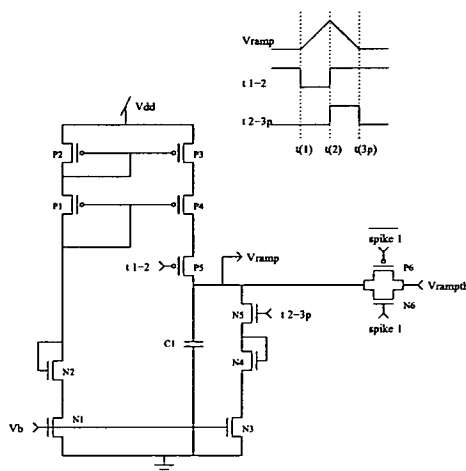


Fig. 2. Current sink/source circuit. C1 charges between $t(1)$ and $t(2)$ and discharges from $t(2)$ to $t(3p)$. The charging and discharging currents are matched to enable prediction of spike 3 at the time when C1 has returned to V_{rampth} .

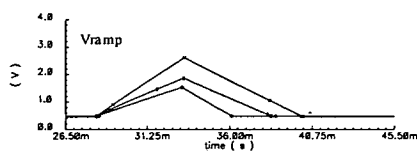


Fig. 3. Simulation outputs for V_{w} . The results of three process mismatch Monte Carlo simulations show a variety of possible shapes for V_{w} .

by the firing times of the input spikes. The timing of this spike can then be compared to that of spike 3 by the spike confirmation block. Spike 3 will only be accepted as genuine if it arrives within a given time window of spike 3p. The adaptive STDP network is also shown. We will attempt to use it to adapt out the process-induced mismatch between the actual firing time $t(3)$ and its predicted time $t(3p)$. The circuit will be described in detail in section 3.

The circuit used to provide the prediction of $t(3)$ is shown in figure 2. Capacitor C1 is charged and discharged by a nominally matched current source and sink. When the voltage across capacitor C1 returns to its initial value a spike is fired, at time $t(3p)$, and if the process were perfect this spike would coincide with the actual spike at $t(3)$. The sequence of events is as follows:

- $t(1)$ Spike 1 is fired and P6 and N6 act as a switch which sets the initial value of V_{ramp} to V_{rampth} .
- $t(1)-t(2)$ P5 is on and C1 is charged through P3 and P4 which supplies a mirrored version of the current set by V_b .
- $t(2)$ P5 turns off while N5 becomes active.

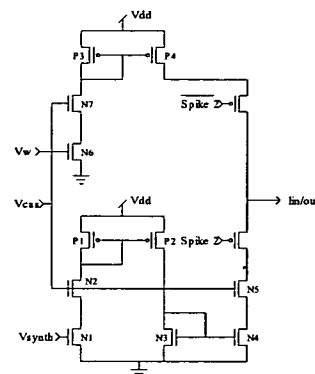


Fig. 4. Excitatory/inhibitory synapse circuit. The synapse is excitatory if V_w is greater than V_{synth} and inhibitory if V_{synth} is greater.

$t(2)-t(3p)$ C1 is discharged through N3 and N4. If the current source and sink are matched the time taken to discharge C1 should match the charging time.

$t(3p)$ V_{ramp} is compared to V_{rampth} using a differential pair. When V_{ramp} crosses it, neuron 3p is "fired". This ends the discharge period.

The accuracy of the prediction depends on how well the current sink and source have been matched. A perfect match is not possible without post fabrication trimming which is expensive and time consuming. Figure 3 is the result of a process mismatch Monte Carlo simulation. The V_{w} signal for three different simulation runs is shown. Clearly, process mismatch causes significant differences not only in the current value but also between pairs of charging and discharging currents leading to an inaccurate prediction. We will use the relationship between the actual occurrence of the spike at $t(3)$ and its corrupted prediction at $t(3p)$ as inputs to the STDP algorithm, aiming to "pull" the actual and predicted spikes back into coincidence. We will, effectively, adapt synaptic delays, as opposed to synaptic strengths, in response to the STDP algorithm. The ability to adapt synaptic delays can be found in other neuromorphic systems which use spike timing [13]-[15].

3. ADAPTIVE CIRCUITRY

To compensate for the mismatch in the current source and sink, at $t(2)$ a current path is opened and charge is injected/removed from the capacitor in figure 2. This effectively increases or decreases the discharge time by raising or lowering the peak signal voltage. The charge is injected/removed by the excitatory/inhibitory synapse designed in [16]. Figure 4 shows the circuit which is controlled by two inputs V_{synth} and V_w . If they are equal the output current is zero. If V_{synth} is greater than V_w the synapse is inhibitory. The synapse is excitatory when V_w rises above V_{synth} . In this application V_{synth} is set to 1.8V. V_{cas} is the bias for the cascoding transistors and is set at 1.37V.

The control of V_w is provided by the circuit in figure 5 which responds to the timing differences between $t(3)$ and $t(3p)$.

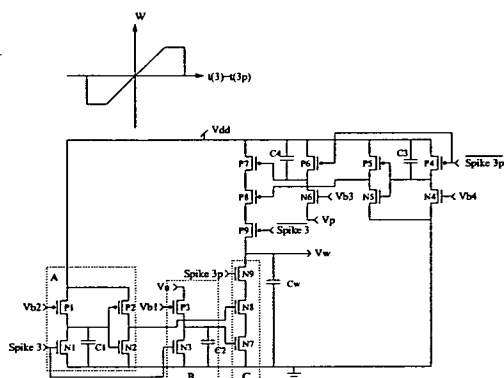


Fig. 5. Weight adaptation circuit and algorithm. V_w is increased if spike 3p occurs before spike 3 and decreased if neuron 3 fires first. Weight change only occurs if spikes 3 and 3p happen within a specified time window. The weight change (ΔW) is “capped” by V_n and V_p to prevent V_w moving from one supply rail to the other. Three circuit blocks are highlighted. A: The circuitry setting the time window. B: The circuitry setting the amount of weight change. C: Combining the two together to create the negative part of the ΔW graph.

The circuit in figure 5 has two distinct parts. MOSFETs N1-3 and P1-3 control reductions in V_w (depression) through N7, N8 and N9, while increases (potentiation) are achieved through transistors N4-6, P4-6, P7, P8 and P9. As the potentiation and depression circuits are mirror-images, with transistor polarities changed, we will present a detailed description of only the depression mechanism.

When spike 3 occurs C1 is discharged through N1 and then charges slowly to Vdd through P1. This results in a pulse at the gate of N8, whose width is determined by V_{b2} . This pulse defines the window within which weight change occurs. At $t(3)$ C2 is discharged through N3 and then charged slowly to V_n through P3. This voltage is N7's gate voltage and controls the size of the weight change. As the voltage across C2 increases with time, the amount of weight change will increase the longer the delay between $t(3)$ and $t(3p)$ until the voltage reaches V_n . This defines the maximum weight change. If spike 3p occurs within the time window, Cw is discharged through N7, N8 and N9 by an amount related to the voltage across C2. When the weight voltage is decreased, the peak of V_{ramp} decreases. This reduces the discharge time and spike 3p will occur at a time $t(3p)$ that is closer to $t(3)$.

The weight change is “capped” by V_n and V_p to prevent V_w moving from one supply rail to the other. The window width is determined empirically from *a priori* knowledge as to when two spikes may be regarded as unrelated events.

When spike 3p is a good prediction of spike 3 no further weight change occurs. Therefore the adaptation network can be left active after calibration. It will then continue to compensate for any circuit drift caused by changes in environmental conditions or charge leakage from the capacitor. The latter is particularly important as many compensation techniques suffer from problems in the per-

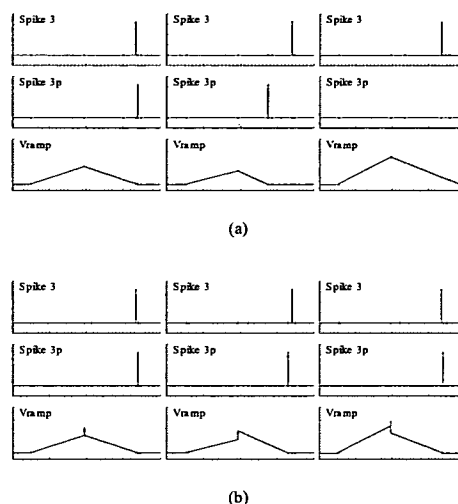


Fig. 6. Pre and post adaptation results. These results show the spike 3, spike 3p and V_{ramp} signals during three process mismatch Monte Carlo simulations. (a) is pre-adaptation and the error in spike 3p is clear. (b) shows the signals post-adaptation where the error has been reduced. The alteration to V_{ramp} can also be seen.

manent storage of calibration values.

4. RESULTS

To test the circuit a 100-run Monte Carlo process mismatch simulation of the AMS 0.35 μ m CMOS CSI process was performed using Cadence Design Tools. During each run the transistor parameters are altered to simulate the process mismatch that occurs during manufacture. Usually this type of analysis is performed to check the robustness of a design but we use it to determine the circuit's ability to *compensate* for mismatch. During each run the difference between $t(3)$ and $t(3p)$ was measured before and after STDP adaptation.

Figure 6(a) shows example results for three runs before adaptation. The timing of spike 3p can be seen to vary widely. After adaptation, figure 6(b), the spread of spikes has been decreased and the peak ramp voltage can be seen to change, on charge injection, to alter the discharge time. The first example is a good match before adaptation and is not affected adversely by it.

While Figure 6(b) is encouraging, a more statistically significant result can be obtained by examining the results for the full 100 runs. Figure 7 shows the spread of the estimation error $t(3) - t(3p)$ before and after STDP adaptation. $t(2) - t(1)$ is 1ms and 5ms in (a) and (b) respectively.

It can be seen very clearly that the standard deviation of the prediction error is reduced dramatically, indicating that prediction after adaptation is far more reliable. The STDP adaptation network has reduced the effect of process mismatch by “pulling together”

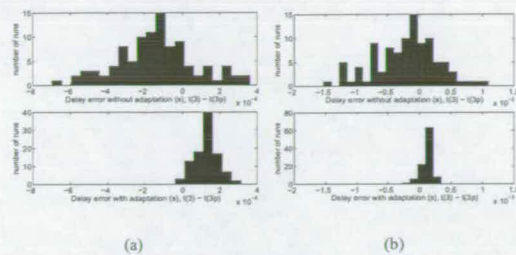


Fig. 7. Distribution of prediction errors. The prediction error is defined as $t(3) - t(3p)$. Each graph has the results before adaptation at the top with after adaptation underneath. Results (a) were from a simulation when $t(2) - t(1) = 1\text{ms}$ and (b) $t(2) - t(1) = 5\text{ms}$

spikes which are effectively coincident in real time, but have been "spread" on silicon by process imperfections.

5. CONCLUSIONS

Simulation results show that spike timing adaptation can minimise the effects of process mismatch in the context of a spike-timing processing scheme for visual scenes.

Our circuit predicts the timing of a third spike from the timing of two previous spikes. This prediction circuit's accuracy depended on the degree of transistor parameter variation. To compensate for this an adaptation network was added to reduce the prediction error.

The circuit was tested using a process mismatch Monte Carlo simulation. Process mismatch degrades the accuracy of prediction significantly. However, STDP adaptation effects a significant and useful reduction in the error spread. The circuit is being fabricated using a CMOS 0.35μ process to verify the simulation results.

It should be noted that this technique differs from dynamic techniques in that it will correct for other imperfections in the circuit, for example variation in neuron firing thresholds or track delay, and not simply mismatch within the current mirror. This technique can also be used when the charging current is a fraction of the discharging current allowing the non-uniform placement of pixels.

Further work will include the integration of the network with a pixel array to demonstrate the potential of a complete system.

The implications of this work are wide-ranging. It is now clear that inter-spike timing can be used both to perform useful processing on silicon and that the same spike timings can drive circuits that adapt to compensate for the unavoidable and ever-increasing mismatches between elements in real silicon chip. This may well prove a major tool in the drive towards reliable computation in an increasingly unreliable medium.

6. REFERENCES

- [1] Marcel J M Pelgrom, Aad C J Duinaijer, and Anton P G Welbers, "Matching properties of MOS transistors," *IEEE*

Journal of Solid-State Circuits, vol. 24, no. 5, pp. 1433–1440, October 1989.

- [2] Eric Vittoz and Jean Fellrath, "CMOS analog integrated circuits based on weak inversion operation," *IEEE Journal of Solid-State Circuits*, vol. SC-12, no. 3, pp. 224–231, June 1977.
- [3] G Wegmann and E A Vittoz, "Basic principles of accurate dynamic current mirrors," *IEE proceedings*, vol. 137, no. 2, pp. 95–100, April 1990.
- [4] J Robert, P Deval, and G Wegmann, "Very accurate current divider," *Electronics Letters*, vol. 25, no. 14, pp. 912–913, 6th July 1989.
- [5] Yen S Yee, Lewis M Terman, and Lawrence G Heller, "A 1 mV MOS comparator," *IEEE Journal of Solid-State Circuits*, vol. SC-13, no. 3, pp. 294–297, June 1978.
- [6] Wulfram Gerstner, "What's different with spiking neurons?," in *Plausible Neural Networks for Biological Modelling*, pp. 23–48. Kluwer Academic Publishers, 2001.
- [7] Philipp Häfner, Misha Mahowald, and Lloyd Watts, "A spike based learning neuron in analog VLSI," in *Advances in Neural Information Processing Systems*, 1996, vol. 9, MIT Press.
- [8] Christal Gordon and Paul Hasler, "Biological learning modeled in an adaptive floating-gate system," in *IEEE International Symposium on Circuits and Systems*, 2002, vol. 5, pp. 609–612.
- [9] Adria Bofill, Alan F Murray, and Damon P Thomson, "Circuits for VLSI implementation of temporally-asymmetric Hebbian learning," in *Advances in Neural Information Processing Systems*, 2002, vol. 14, MIT Press.
- [10] Adria Bofill-i-Petit and Alan F Murray, "Learning temporal correlations in biologically-inspired aVLSI," in *IEEE International Symposium on Circuits and Systems*, May 25–28 2003, vol. 5, pp. 817–820.
- [11] Adria Bofill-i-Petit and Alan F Murray, "Synchrony detection by analogue VLSI neurons with bimodal STDP synapses," in *Advances in Neural Information Processing Systems*, December 2003, Accepted for publication.
- [12] F Wörgötter, A Cozzi, and V Gerdes, "A parallel noise-robust algorithm to recover depth information from radial flow fields," *Neural Computation*, vol. 11, pp. 381–416, 1999.
- [13] Wulfram Gerstner, Richard Kempter, J Leo van Hemmen, and Hermann Wagner, "Hebbian learning of pulse timing in the barn owl auditory system," in *Pulsed Neural Networks*, pp. 353–377. MIT Press, 1998.
- [14] Harald Hünig, Helmut Glünder, and Günther Palm, "Synaptic delay learning in pulse-coupled neurons," *Neural Computation*, vol. 10, pp. 555–565, 1998.
- [15] Christian W Eurich, Klaus Pawelzik, Udo Ernst, Jack D Cowan, and John G Milton, "Dynamics of self-organized delay adaptation," *Physical Review Letters*, vol. 82, no. 7, pp. 1594–1597, 15th February 1999.
- [16] Sebastian Loeda, *Circuits to Support Antisymmetric Hebbian Training in Artificial Neural Networks*, BEng Hons Thesis, The University of Edinburgh, May 2002.

COMPENSATING MISMATCH IN A DEDICATED PIXEL ARRAY FOR MOVING EDGE DETECTION

Vasin Boonsobhak, Katherine Cameron, David Renshaw and Alan Murray

School of Engineering and Electronics
The University of Edinburgh
Edinburgh, EH9 3JL, UK

Email: {v.boonsobhak,k.l.cameron,d.renshaw,a.f.murray}@ed.ac.uk

ABSTRACT

This paper describes the implementation of a CMOS edge-detecting pixel array with on-chip adaptation that compensates for intrinsic variation between devices. The adaptation algorithm, based on asymmetric Hebbian training, uses spike timing to adapt out the effects of device mismatch and process variation. Each pixel adapts to background illuminance and provides high-pass filtering output with respect to local positive and negative illuminance transients. Sample chips have been manufactured using a $0.35\mu\text{m}$ CMOS technology, and the test results confirming the feasibility of the chosen approach are reported.

1. INTRODUCTION

The continuously expanding field of machine vision, e.g., automotive applications, demands for high-performance and low-cost image sensors and system implementations. CMOS technology allows to integrate sophisticated sensor technology with massively parallel image and signal processing. This motivated a plethora of image sensor implementations including designs with integrated preprocessing [1, 2, 3]. The high density of visual information in our environment makes real-time image processing a major challenge, in spite of the ever increasing speed of available electronic processing circuits. However, image data tends to be highly redundant and thus compressible without information loss. A very basic and useful image compression function is the enhancement of local brightness transients [4, 5].

Our research was motivated by the objective to develop an image sensor capable of recovering depth information. Our exemplar system was developed by Wörgötter *et al* [6]. It is a spike-timing driven algorithm for visual scene analysis whose temporal characteristics will be degraded if implemented in analogue focal-plane VLSI. We have based our pixel upon the light transducing elements described in [7] and [8]. The technique for device mismatch correction is based upon a form of asymmetric Hebbian learning, a neural algorithm implementing Spike Timing Dependent Plasticity (STDP). This class of algorithms, driven not

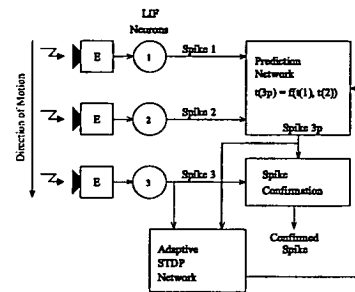


Fig. 1. System block diagram. Neurons 1 to 3 fire when an edge passes the corresponding pixels. Neuron 3p fires at a time predicted from the firing time of neurons 1 and 2. If the neurons fire at equal intervals, $t(3p) = t(2) + \{t(2) - t(1)\}$ where $t(1)$ is the firing time of spike 1, $t(2)$ for spike 2 etc. Spike 3p is the prediction of spike 3.

by spike-rate correlations but by inter-spike timings, has been shown to recover different and potentially richer information from a spike train when compared to rate and population coding methods [9]. STDP is a relatively simple algorithm which has been successfully implemented in analogue VLSI [10, 11, 12].

The system and STDP circuits are presented in section 2. The edge-detecting pixel is described in section 3. In section 4, experimental results demonstrate both the problems caused by mismatch and the benefits of the adaptation network.

2. SPIKE-TIMING DRIVEN CIRCUITS

The system block diagram is shown in Fig. 1. The spike firing circuit comprises three leaky integrate-and-fire (LIF) neurons [11]. These LIF neurons integrate output currents received from edge-detecting pixels (E) and fire when the accumulated signal reaches the threshold. The spikes from neurons 1 and 2 are passed to the prediction network which generates another spike, spike 3p (i.e. "predicted", not actual), at a time determined by the firing times of the input spikes. The timing of spike 3p can then be compared to that of spike 3 by the spike confirmation block. Spike 3 will only be accepted as genuine if it arrives within a

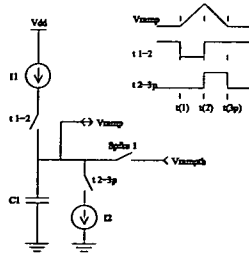


Fig. 2. Current sink/source circuit. C_1 charges between $t(1)$ and $t(2)$ and discharges from $t(2)$ to $t(3p)$. The charging and discharging currents are matched to enable prediction of spike 3 at the time when C_1 has returned to V_{rampth} .

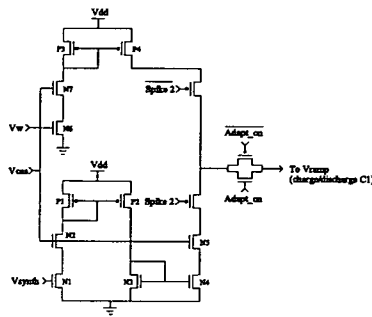


Fig. 3. Excitatory/inhibitory synapse circuit. The synapse is excitatory if V_w is greater than V_{synth} and inhibitory if V_{synth} is greater.

given time window of spike 3p. The adaptive STD network is also shown. We attempt to use it to adapt out the process-induced mismatch between the actual firing time $t(3)$ and its predicted time $t(3p)$.

2.1. Prediction circuitry

The circuit used to provide the prediction of $t(3)$ is shown in Fig. 2. Capacitor C_1 is charged and discharged by a nominally matched current source and sink (I_1, I_2). When the voltage across capacitor C_1 returns to its initial value a spike is fired, at time $t(3p)$, and if the process were perfect this spike would coincide with the actual spike at time $t(3)$. This circuit has been described in [13].

2.2. Adaptive circuitry

The accuracy of the prediction depends on how well the current source and sink (I_1, I_2) have been matched. To compensate for the mismatch in I_1 and I_2 , at time $t(2)$ a current path is opened and charge is injected or removed from the capacitor C_1 in Fig. 2. This effectively increases or decreases the discharge time by raising or lowering the peak of V_{rampth} .

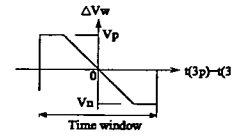


Fig. 4. Weight-change curve. V_w is increased if spike 3p occurs before spike 3 and decreased if spike 3 is fired first.

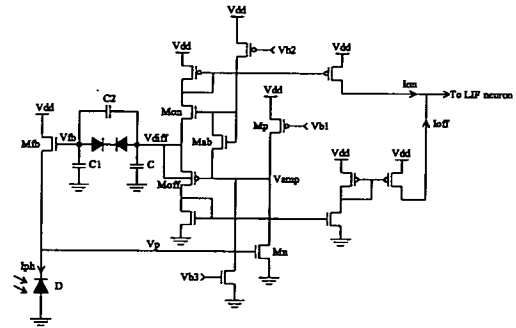


Fig. 5. Circuit diagram of the edge-detecting pixel. Positive (ON) and negative (OFF) transients are combined on a single output node and fed to a neuron.

The charge is injected or removed by the excitatory/inhibitory synapse designed in [14]. Fig. 3 shows the circuit which is controlled by two inputs V_{synth} and the weight voltage V_w . If they are equal, the output current is zero. If V_{synth} is greater than V_w , the synapse is inhibitory. The synapse is excitatory when V_w rises above V_{synth} . V_{cas} is the bias for the cascoding transistors.

The control of V_w , based on asymmetric Hebbian learning, is shown in Fig. 4. The weight-change curve responds to the timing differences between $t(3)$ and $t(3p)$. Weight change ΔV_w , restricted by V_n and V_p to prevent V_w moving from one supply rail to the other, only occurs if spikes 3 and 3p happen within a specified time window. The window width is determined empirically from *a priori* knowledge as to when two spikes may be regarded as unrelated events. Detailed description of the weight adaptation circuit has been given in [13].

3. EDGE-DETECTING PIXEL

Circuit diagram of the edge-detecting pixel is shown in Fig. 5. The circuit combines an adaptive photoreceptor [7] with a rectifying differentiating element [8]. It consists of a photodiode D in series with a transistor M_{fb} in source-follower configuration, and a negative feedback loop is created between the source and the gate of M_{fb} . The feedback loop consists of a high-gain inverting amplifier in common-source configuration (M_n, M_p) and a rectifying temporal class-AB differentiator stage ($M_{on}, M_{off}, M_{ab}, C$). In con-

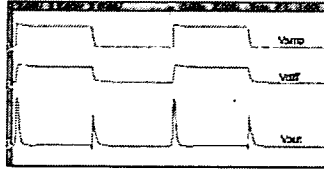


Fig. 6. Irradiance step response of the edge-detecting pixel. These oscilloscope traces show V_{amp} , V_{diff} and $V_{out} = 1M\Omega \times (I_{on} + I_{off})$. V_{out} trace is the temporal derivative of the input contrast step.

trast to [7], we employ an Nwell-to-Psubstrate photodiode as the light transducer because it has reasonably high quantum efficiency, small parasitic capacitance and low dark current [15].

An additional capacitive gain stage in the feedback loop gives enhancement to the response [7]. The voltage variations on the differentiator node V_{diff} are amplified with respect to the variations of V_{fb} by the capacitive divider ratio $A_C = (C_1 + C_2)/C_2$ and so are the transient currents I_{on} and I_{off} .

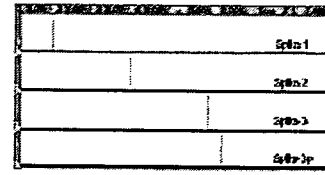
The resistive element, constructed from two oppositely directed diodes in series, ensures that V_{diff} eventually adapts to a dc value close to V_{fb} . In the ideal case, the current is limited to a low value by the reverse-biased diode, such that the element exhibits a symmetric saturating sigmoidal current-voltage characteristic.

4. EXPERIMENTAL RESULTS

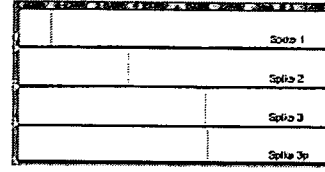
The measured responses of the edge-detecting pixel is shown in Fig. 6. The measurement was done by illuminating the entire chip through a diffuser with a radiance-modulated light-emitting diode (LED), operating around an emission wavelength of 590 nm, while shielding the chip from ambient light. The power supply voltage was set to 3.3 V and the bias voltages V_{b1} , V_{b2} and V_{b3} were fixed for all measurements such that M_p and M_n were operated slightly above threshold. A radiance-voltage calibration was performed using a digital light meter.

As the pixel provides output in the form of current $I_{on} + I_{off}$, a $1M\Omega$ resistor was connected between the output node and ground, enabling voltage readout for an oscilloscope. As a result, the output time constant was massively increased and does not reflect the natural on-chip response of the pixel. The circuit was allowed to reach a steady state at a default illuminance of 1 lux. Saturation and adaptation effects can be observed in the V_{amp} and V_{diff} traces after high-contrast step of 100 lux.

The ability to adapt to process mismatch was tested by stimulating the combined pixel/STDP circuit with a moving light pattern generated by an array of flashing LEDs (10 lux). The LED test pattern was set in such a way that edge-detecting pixels were illuminated chronologically. The optical part of the imaging sys-



(a)



(b)

Fig. 7. Pre and post adaptation waveforms. These oscilloscope traces show the spike 1, spike 2, spike 3 and spike 3p measured during experiments when $t(2) - t(1) = 5ms$. (a) is pre-adaptation and the error in spike 3p is clear. (b) shows the post-adaptation signals where the error has been reduced.

tem was a surveillance camera lens with a focal length of 6 mm and an f -number of 1.2. The LED array was placed at a distance of 15 cm from the lens. The experiments were repeated 10 times over five chips and conducted under ambient lighting condition of ac-driven fluorescent office lighting (150 lux).

Fig. 7 shows an example of spike waveforms observed in the experiments before and after STDP adaptation. It is important to note that each pixel is able to detect the LED test pattern (high-frequency signal) while rejecting the ambient light (low-frequency signal). Fig. 8 shows the spread of the prediction error $t(3p) - t(3)$ before and after STDP adaptation. It can be seen that, after adaptation, the prediction error is reduced by at least 5 times. The prediction error when $t(2) - t(1) = 5ms$ is always greater than the error when $t(2) - t(1) = 1ms$ because it is related to the mismatch in current and the charging time. The STDP adaptation network has reduced the effect of process mismatch by “pulling together” spikes which are effectively coincident in real time, but have been spread on silicon by process imperfections.

5. CONCLUSION

An edge-detecting pixel array with on-chip adaptation that compensates for device mismatch has been presented and characterised. Our first chip was manufactured using a $0.35\mu m$ CMOS technology. Experimental results show that the pixels adapt to background illuminance and provide high-pass filtering output, and that spike timing adaptation can minimise the effects of process mismatch in the context of a spike-timing processing scheme for visual scene.

In future work, this approach will be extended to include the depth-from-motion algorithm. Our long term

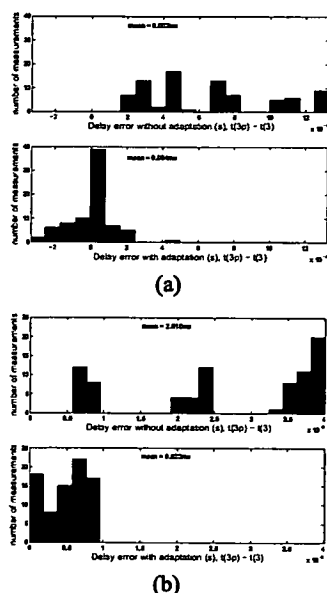


Fig. 8. Distribution of prediction errors. Results (a) were from the measurement when $t(2) - t(1) = 1\text{ms}$ and (b) $t(2) - t(1) = 5\text{ms}$.

goal is to achieve a specialised silicon retina that provides locally processed depth information, which either can be used for certain application segments, or which can serve as a building block for an intelligent application-specific chip in machine vision.

6. ACKNOWLEDGEMENTS

The first author is grateful to Mahanakorn University of Technology, Thailand for financial support.

7. REFERENCES

- [1] S.G. Smith, J.E.D. Hurwitz, M.J. Torrie, D.J. Baxter, A.A. Murray, P. Likoudis, A.J. Holmes, M.J. Panaghiston, R.K. Henderson, S. Anderson, P.B. Denyer, and D. Renshaw, "A single chip CMOS 306×244 pixel NTSC video camera and a descendant coprocessor device," *IEEE J. Solid-State Circuits*, vol. 33, no. 12, pp. 2104–2111, December 1998.
- [2] M. Loose, K. Meier, and J. Schemmel, "A self-calibrating single-chip CMOS camera with logarithmic response," *IEEE J. Solid-State Circuits*, vol. 36, no. 4, pp. 586–596, April 2001.
- [3] K. Yoon, C. Kim, B. Lee, and D. Lee, "Single-chip CMOS image sensor for mobile applications," *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1839–1845, December 2002.
- [4] Y. Muramatsu, S. Kurosawa, M. Furumiya, H. Ohkubo, and Y. Nakashiba, "A signal-processing CMOS image sensor using a simple analog operation," *IEEE J. Solid-State Circuits*, vol. 38, no. 1, pp. 101–106, January 2003.
- [5] S. Mizuno, K. Fujita, H. Yamamoto, N. Mukozaka, and H. Toyoda, "A 256×256 compact CMOS image sensor with on-chip motion detection function," *IEEE J. Solid-State Circuits*, vol. 38, no. 6, pp. 1072–1075, June 2003.
- [6] F. Wörgötter, A. Cozzi, and V. Gerdes, "A parallel noise-robust algorithm to recover depth information from radial flow fields," *Neural Computation*, vol. 11, pp. 381–416, 1999.
- [7] T. Delbrück and C.A. Mead, "Analog VLSI adaptive logarithmic wide-dynamic-range photoreceptor," in *Proc. 1994 IEEE Int. Symp. on Circuits and Systems*, London, UK, May 1994, vol. 4, pp. 339–342.
- [8] J. Kramer, "An integrated optical transient sensor," *IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 49, no. 9, pp. 612–628, September 2002.
- [9] W. Gerstner, "What's different with spiking neurons?," in *Plausible Neural Networks for Biological Modelling*, pp. 23–48. Kluwer Academic Publishers, 2001.
- [10] P. Häfziger, M. Mahowald, and L. Watts, "A spike based learning neuron in analog VLSI," in *Advances in Neural Information Processing Systems*, vol. 9. MIT Press, 1996.
- [11] A. Bofill-i-Petit and A.F. Murray, "Learning temporal correlations in biologically-inspired aVLSI," in *Proc. 2003 IEEE Int. Symp. on Circuits and Systems*, Bangkok, Thailand, May 2003, vol. 5, pp. 817–820.
- [12] A. Bofill-i-Petit and A.F. Murray, "Synchrony detection by analogue VLSI neurons with bimodal STDP synapses," in *Advances in Neural Information Processing Systems*, vol. 16. MIT Press, 2004.
- [13] K.L. Cameron and A.F. Murray, "Can spike timing dependent plasticity compensate for process mismatch in neuromorphic analogue VLSI?," in *Proc. 2004 IEEE Int. Symp. on Circuits and Systems*, Vancouver, Canada, May 2004, vol. 5, pp. 748–751.
- [14] S. Loeda, *Circuits to Support Antisymmetric Hebbian Training in Artificial Neural Networks*, B.Eng. Hons. Thesis, May 2002, The University of Edinburgh.
- [15] J. Goy, B. Courtois, J.M. Karam, and F. Pressecq, "Design of an APS CMOS image sensor for low light level applications using standard CMOS technology," *Analog Integrated Circuits and Signal Processing*, vol. 29, pp. 95–104, 2001.

Spike Timing Dependent Adaptation for Mismatch Compensation

Katherine Cameron, Alan Murray
School of Engineering and Electronics
The University of Edinburgh
Edinburgh, EH9 3JL, UK
Email: k.cameron@ed.ac.uk, alan.murray@ee.ed.ac.uk

Steve Collins
Department of Engineering Science
University of Oxford
Oxford, OX1 3PJ, UK
Email: steve.collins@eng.ox.ac.uk

Abstract—This paper presents some circuitry for use within a visual-processing depth-recovery algorithm based upon spike timing. The accuracy of the depth calculation relies on a prediction which, when implemented in analogue VLSI, will be degraded by transistor mismatch. An adaptive circuit based on Spike Timing Dependent Plasticity (STDP) was designed to reduce the effects of VLSI process variations on the algorithm's performance. Simulation Results for the circuitry, designed using a 0.35 μ m process, are reported.

I. INTRODUCTION

Spike Timing Dependent Plasticity (STDP) is a neural algorithm operating not on spike-rate correlations but individual inter-spike timings. STDP therefore has the ability to adapt synaptic delays [1]-[3] to improve or "tune" network performance. The interesting properties of STDP have been investigated and successfully implemented in analogue VLSI [4]-[8]. Furthermore, the Monte Carlo simulation results reported in this paper come from a design which is an extension of our previous work, fabricated in 0.35 μ m silicon [9], which showed the ability to correct for mismatch effects.

The problems of transistor mismatch are exacerbated as device sizes are driven further into sub-micron geometries [10]. With an increasing number of devices on chip the demand for lower power consumption encourages sub-threshold design where the problem of mismatch is even greater [11]. As mismatch can effect circuit time constants, we have chosen to exploit STDP's ability to adapt delays and investigate the use of a time dependent spiking algorithm which compensates for the mismatch error within an exemplar neural system.

This paper will present the circuitry designed and Monte Carlo simulation of the circuit extraction results showing both the problems caused by mismatch and the benefits of the adaptation network. The circuitry will be fabricated using a 0.35 μ m process to verify the simulation results. Although these results are based upon an explicitly spike-driven depth-from motion algorithm, the techniques demonstrated are generic to any algorithm that is based upon inter-spike correlations. As a result, the results herein and the chip that implements the STDP process do not have a specific application. Rather, they aim to develop new forms of robust circuit for the noisy, mismatched and unreliable building nanoscale building blocks of future chips.

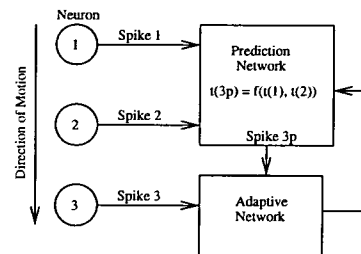


Fig. 1. System Block Diagram. Neurons 1 to 3 fire at predetermined intervals. Neuron 3p fires at a time predicted from the firing time of neurons 1 and 2. If the neurons fire at equal intervals, $t(3p) = t(2) + \{t(2) - t(1)\}$ where $t(1)$ is the firing time of spike 1, $t(2)$ for spike 2 etc. Spike 3p is the prediction of spike 3.

II. PREDICTIVE CIRCUITRY

We chose the vision-processing system described by Wörgötter et al. [12] to show STDP's mismatch compensation ability. It is a depth-from-motion algorithm which uses relative neural spike timings to recover depth information from radial flow-fields. The algorithm was found to be susceptible to noise and a prediction mechanism was added to make it more robust. The prediction uses the timing of two spikes to determine when a third spike will arrive. If the spike arrives within a small time window of the prediction it is treated as genuine. Timing mismatches are therefore potentially very damaging to this algorithm, rendering it a good candidate for this study.

We used a prediction network (figure 1) designed for the condition where the time between spikes 1 and 2 was equal to that between spikes 2 and 3. In this case the time of the predicted spike 3, spike 3p, is $t(2) + \{t(2) - t(1)\}$ where $t(1)$ is the time of spike 1, $t(2)$ is the time of spike 2 etc.

Preliminary attempts to compensate for mismatch error are described in [9]. The original design suffered from the problem that most of the mismatch was caused by a gain error, i.e. differing g_{ms} , but no gain correction could be applied. Instead an offset correction, equivalent to addition or subtraction of time, was applied which was only applicable to a fixed time delay, in that case 5ms. When the delay was changed to 1ms the circuit had to adapt to a different level of correction.

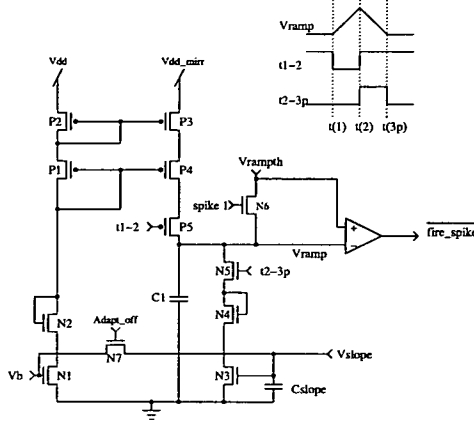


Fig. 2. Spike prediction circuit. C1 charges between $t(1)$ and $t(2)$ and discharges from $t(2)$ to $t(3p)$. If N7 is closed V_b equals V_{slope} and an accurate prediction will be given if the transistors have good matching. If that is not the case N7 can be opened and V_{slope} can be set at a different voltage to ensure equal charge and discharge times.

The predictive circuitry was re-designed to address this issue and is shown in figure 2. The sequence of operation is:

- $t(1)$ Spike 1 is fired and N6 acts as a switch which sets the initial value of V_{ramp} to V_{rampth} .
- $t(1)-t(2)$ P5 is on and C1 is charged through P3 and P4 which supplies a mirrored version of the current set by V_b .
- $t(2)$ P5 turns off while N5 becomes active.
- $t(2)-t(3p)$ C1 is discharged through N3 and N4 at a rate set by the voltage across C_{slope} . If the current source and sink are matched and V_b equals V_{slope} the time taken to discharge C1 should match the charging time.
- $t(3p)$ V_{ramp} is compared to V_{rampth} using a differential pair. When V_{ramp} crosses it, neuron 3p is "fired". This ends the discharge period.

The bias voltage applied to N1 is now no longer permanently connected to N3. The closing of N7 allows the voltage across C_{slope} to be initialised to V_b but it can be open to enable different bias voltages to be set. The W/L ratio for transistors N3 and N4, and therefore N1 and N2, was altered to reflect the driving of N3 by the voltage across C_{slope} . The length of the transistors determines their response to the changing voltage V_{ramp} and was set at $3\mu m$. To reduce the capacitive coupling of V_{ramp} to V_{slope} it was important to keep the gate-drain parasitic capacitance to a minimum. The width chosen to reduce this effect was $0.6\mu m$.

If N7 is closed the prediction accuracy demands that the matching between transistors be good. Should mismatch occur, the discharging current can be greater or less than the charging current resulting in an early or late prediction. The mismatch could be eliminated by post-fabrication trimming but this

is expensive and time-consuming. Instead we minimise the effect of the mismatch by altering V_{slope} . The alteration is performed by the adaptive circuitry described in the next section based on the time difference between spikes 3 and 3p. Effectively, we use STDP to restore the near-perfect synchrony that has been corrupted by mismatch.

III. ADAPTIVE CIRCUIT DETAILS

Transistors N1 and N3 in figure 2 have sub-threshold gate voltages so any change in V_{slope} could potentially have a large impact on the discharging current. To ensure a small voltage change a switched capacitor technique is used as illustrated in figure 3. The HiLo circuit uses source followers to generate signals approximately 200mV above and below V_{slope} . If spike 3 arrives before the prediction, V_{slope} must be increased and the up signals, up , up_1 and up_{1d} , are active. If spike 3p arrives first, the $down$ signals are active but the operation is essentially the same. Therefore only the process of increasing V_{slope} is described here.

A capacitor bank, C0-3, is used to implement the change in V_{slope} in a time dependent manner, as opposed to merely direction dependent. It is used in a similar way to that described in [13]. If a single capacitor were to be used then the change in V_{slope} would always be by a fixed amount, either up or down, irrespective of the accuracy of the prediction. This would lead to a choice between either very slow convergence and high accuracy or fast convergence with low accuracy. The inclusion of the capacitors is controlled by $CS1-3$ which are reset when the up signal is generated at the arrival of spike 3. The NOR and NOT gates marked S are "current starved", by a sub-threshold biased transistor connected between VDD and the standard circuit configuration, and therefore have a slow rise time. After the up signal, the output of the NOR gate

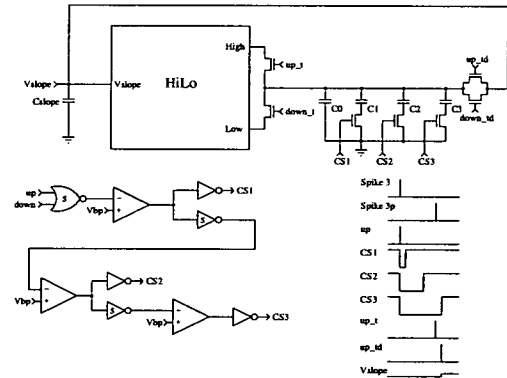


Fig. 3. V_{slope} Update circuitry. The circuit employs a switched capacitor technique to increase or decrease the voltage V_{slope} . The amount of change is determined by the signals $CS1-3$ which turn on over time, one after another. The logic gates marked S have a slow rise time and implement the delay between CS signals. The HiLo circuitry uses source followers to generate signals approximately $\pm 200mV$ around V_{slope} .

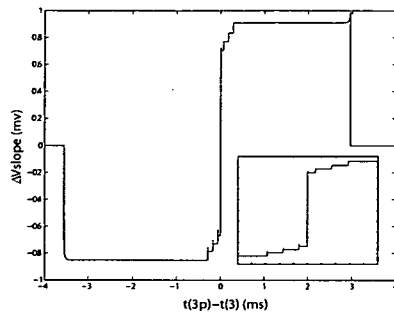


Fig. 4. The change in V_{slope} vs prediction error. The change in voltage of V_{slope} implemented by the circuitry shown in figure 3. The inset graph is of the range $-500\mu s - 500\mu s$ and shows the effect of switching in the capacitors C1-3 in figure 3.

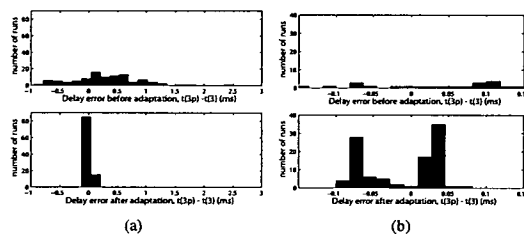


Fig. 5. Distribution of prediction errors. The time difference, $t(3p) - t(3)$, was measured before and after adaptation. The x-axis scale is set using the pre-adaptation range in (a) and the post-adaptation range in (b).

slowly rises until it reaches V_{bp} , in this case 2.5V. At this point $CS1$ becomes high and the output of the first slow NOT gate starts to rise. With $CS1$ high, charge can be stored on the connected capacitor. $CS2$ and $CS3$ become high approximately $100\mu s$ after the preceding CS signal. When spike $3p$ arrives up_{\perp} is generated and the *High* voltage from the *HiLo* circuit is stored across all active capacitors. Therefore the amount of stored charge is related to the time between spike 3 and spike $3p$. Up_{\perp} is generated slightly later at which point the charge is shared between the capacitor bank and C_{slope} resulting in a rise in V_{slope} . C_{slope} is a much larger capacitor than the others therefore the voltage change is much smaller than the 200mV difference generated by the *HiLo* circuitry. Figure 4 shows the resulting change of V_{slope} with different time delays between the actual and predicted spikes. A windowing mechanism is used so that if a predicted spike arrives much earlier or later than the actual spike they are deemed to be unrelated. This results in the zero change at $|t(3p) - t(3)| > 3ms$. As the adaptive network drives V_{slope} towards a particular point it can be left active after calibration. It will then continue to compensate for any circuit drift caused by changes in environmental conditions or charge leakage from the capacitor. The only constraint is that more correct spikes are present than

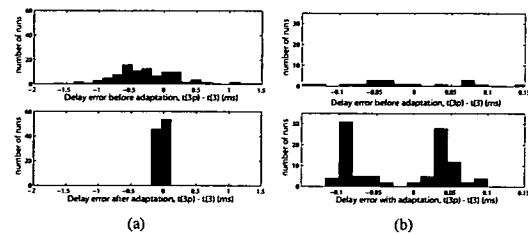


Fig. 6. Distribution of prediction errors when $t(2) - t(1) = \frac{5}{6} \{t(3) - t(2)\}$. The time difference, $t(3p) - t(3)$, was measured before and after adaptation. The x-axis scale is set using the pre-adaptation range in (a) and the post-adaptation range in (b).

incorrect ones.

IV. RESULTS

The circuitry was designed using the AMS $0.35\mu m$ C35 process. The testing was performed using Monte Carlo mismatch simulations of the circuit extraction in which the transistor and capacitor characteristics span the measured statistics of the fabrication process.

100 Monte Carlo simulations were run with a time delay between spikes of 5ms. The time difference, $t(3p) - t(3)$, was measured before and after adaptation. The results are shown in figure 5(a) and the improvement in the prediction is clear. Figure 5(b) shows the same data but this time the x-axis scale has been set to reflect the post-adaptation error range.

It can be seen that very few pre-adaptation results are as good as the full range of post-adaptation results. It is also a significant improvement on the results from our previous design reported in [9]. The two separate distributions visible in (b) are caused by the shape of the voltage change curve in figure 4. Around $t(3p) - t(3) = 0$ there is a swing of approximately 1.4mV. This means that when the prediction is close to $t(3)$ it tends to oscillate between “too fast” and “too slow” but is unable to get exactly to the zero point.

It is also possible to predict a time proportionate but not equal to $t(2) - t(1)$ by ratioing the widths of transistors N1 and N3 in figure 2. We plan to fabricate a prediction circuit with a 5:6 ratio, i.e a 5ms time delay will result in a prediction of 6ms, and the results of the extracted simulation are shown in figure 6.

While improving the prediction quality is important it was also an aim of this work to design a correction method that would work while the time delay between spikes 1 and 2 was varying, as it would when objects at different depths are being processed. Two spike trains were applied to both this circuit and the previous design [9] during Monte Carlo simulations. The first train had 25 sets of spikes and the time delay varied between 3 and 6ms. A set is defined as a spike 1, spike 2, spike 3 group. The second was a longer train of 33 sets with a larger delay range of 1 to 6ms. The time difference between every pair of spikes 3 and $3p$ was measured and the results of 5 simulations are shown in figure 7. In both cases the initial

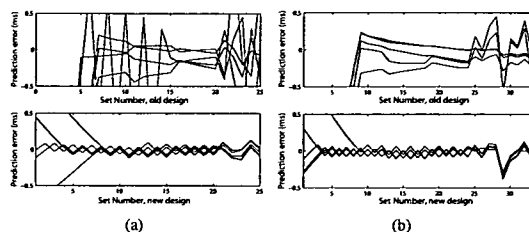


Fig. 7. Prediction error during adaptation. Spike trains were applied during 5 Monte Carlo simulations with varying time delays to the current circuitry and a previous design. (a) 25 spike sets were applied over a range of 3 - 6ms. (b) 33 spike sets were applied over a range of 1 - 6ms.

applied delays varied only a small amount around 5ms with the larger swings at the end of the simulation. Note also that the initial adaptation of the new circuitry can be seen as the error converges around 0. In figure 7(a) it is clear that the new circuitry provides a consistently good prediction error as the delay time changes. This is a significant improvement when compared to the results for the previous design. In (b) there is a less dramatic improvement but it is still quite clear. For the new design the only time an error is registered outside of the range $\pm 150\mu\text{s}$ is when the delay jumps from 1 to 6ms and the error has recovered by the next iteration. The difference between the results is partly due to the increased range of time delays and the number of spike sets applied. The 25 set simulation did not allow the original design to reach a particularly stable point before the delay began to vary. In one case this even resulted in a situation where no convergence took place. The 33 set simulation gave slightly more stability to the input and the old design results were better. While the gain correction provides a better result over changing time delays than the offset correction it is still not perfect. This is the result of the error in the prediction being composed of both a gain and an offset error.

V. CONCLUSION

We have shown through simulation that spike timing dependent algorithms can be used successfully to reduce the effects of device mismatch within the context of a neural processing system for visual scenes.

The circuitry generates a prediction for spike arrival from the timing of two previous spikes. This prediction can be degraded by device mismatch and the temporal difference between the prediction and actual spike is used to adjust circuit parameters until the prediction is accurate. The Monte Carlo simulation results indicate a superior performance under steady state input when compared to our previous design. It also solves much of the problem when the input has a varying time delay.

This result demonstrates a new use of spike-timing as a route to the production of reliable neural architectures implemented on "unreliable" substrates. We use the knowledge that timing relationships are present in input signals from

sensors to perform unsupervised, autonomous calibration of timing circuitry. We are led to suggest that the technique has wider implications for other forms of sensor and for all spiking neural systems fabricated in real, imperfect technologies. It demonstrates yet another useful feature of spike-based coding and may well be part of the evolutionary advantage exploited by the nervous system in its use of spiking neurons as a computational substrate.

As indicated in the introduction, these techniques are generic to any algorithms that use inter-spike correlations. In more general spike-driven architectures, where explicit correlations are not part of the process, correlated spike trains can be introduced to the network in an initial calibration phase, within which autonomous self-calibration occurs. Approaches of this nature will be vital to robust systems using deep-sub-micron devices, to avoid a heavily-supervised calibration methodology that will be both time-consuming and a waste of circuit area.

ACKNOWLEDGMENTS

The authors wish to acknowledge the helpful discussions with Vasin Boonsobhak.

REFERENCES

- [1] W. Gerstner, R. Kempter, J. L. van Hemmen, and H. Wagner, "Hebbian learning of pulse timing in the barn owl auditory system," in *Pulsed Neural Networks*, W. Maass and C. M. Bishop, Eds. MIT Press, 1998, ch. 14, pp. 353-377.
- [2] H. Hünig, H. Glünder, and G. Palm, "Synaptic delay learning in pulse-coupled neurons," *Neural Computation*, vol. 10, pp. 555-565, 1998.
- [3] C. W. Eurich, K. Pawelzik, U. Ernst, J. D. Cowan, and J. G. Milton, "Dynamics of self-organized delay adaptation," *Physical Review Letters*, vol. 82, no. 7, pp. 1594-1597, 15th February 1999.
- [4] A. Bofill-i-Petit and A. F. Murray, "Learning temporal correlations in biologically-inspired aVLSI," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 5, May 25-28 2003, pp. 817-820.
- [5] G. Indiveri, "Neuromorphic bistable VLSI synapse with spike-timing-dependent plasticity," in *Advances in Neural Information Processing Systems*, vol. 15, 2003, pp. 1091-1098.
- [6] A. Bofill-i-Petit and A. F. Murray, "Synchrony detection by analogue VLSI neurons with bimodal STDP synapses," in *Advances in Neural Information Processing Systems*, vol. 16, MIT Press, 2003.
- [7] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI reconfigurable network of integrate-and-fire neurons with spike-based learning synapses," in *Proc. European Symposium on Artificial Neural Networks*, April 28-30 2004, pp. 405-410.
- [8] A. Bofill-i-Petit and A. F. Murray, "Synchrony detection and amplification by silicon neurons with STDP synapses," *IEEE Trans. Neural Networks*, vol. 5, no. 15, pp. 1296-1304, September 2004.
- [9] K. Cameron, V. Boonsobhak, A. Murray, and D. Renshaw, "Spike timing dependent plasticity (STDP) can ameliorate process variations in neuromorphic VLSI," *IEEE Trans. Neural Networks*, Accepted for Publication 2005, available at <http://www.see.ed.ac.uk/~s9727694/TNN0705.pdf>.
- [10] M. J. M. Pelgrom, A. C. J. Duinaijer, and A. P. G. Welbers, "Matching properties of MOS transistors," *IEEE J. Solid-State Circuits*, vol. 24, no. 5, pp. 1433-1440, October 1989.
- [11] F. Forti and M. E. Wright, "Measurement of MOS current in the weak inversion region," *IEEE J. Solid-State Circuits*, vol. 29, no. 2, pp. 138-142, February 1994.
- [12] F. Wörgötter, A. Cozzi, and V. Gerdes, "A parallel noise-robust algorithm to recover depth information from radial flow fields," *Neural Computation*, vol. 11, pp. 381-416, 1999.
- [13] R. J. Vogelstein, U. Mallik, and G. Cauwenberghs, "Silicon spike-based synaptic array and address-event transceiver," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. V, 2004, pp. 385-388.

A Cognitive Vision Model with STDP Adaptation

Zhijun Yang, Katherine Cameron, Vasin Boonsobhak and Alan Murray

School of Engineering & Electronics, Edinburgh University, Mayfield Rd., Edinburgh, EH9 3JL, UK

Zhijun.Yang@ed.ac.uk, K.L.Cameron@sms.ed.ac.uk, V.Boonsobhak@ed.ac.uk, Alan.Murray@ee.ed.ac.uk

Early cognitive vision is one of the most important perceptive functions in the mammalian cerebral cortex. During information processing in the visual cortex, many bioelectrical signals are exchanged between cortical neurons, across their synaptic interconnections. When modelling visual cortex processing a dynamic scene can be modelled as an optical flow field, which can, in turn, be mapped on to a neuronal network. A possible, simple network structure has neurons placed along axes arranged radially from the optical flow field centre with only nearest-neighbour, on-axis connections. The characteristics of a dynamic scene can thus be reconstructed by local computation in neurons based upon the optical flow field.

We have modelled a large-scale leaky integrate-and-fire (LIF) neuronal network in precisely this network arrangement. The model uses edge-sensitive pixels to capture the position and speed of movement of features in a moving scene. Subsequently, object identification is achieved and depth information is recovered. In this example, we have restricted movement to a straight line, with a constant relative speed between a moving cognitive model and the static scene. Each neuron has two excitatory input synapses. One connects to the edge-sensitive pixel and is referred to as the receptive synapse. The second receives a prediction of the time-of-arrival of an edge from the previous neuron in the flow field, and is referred to as the flow synapse. When an edge passes a pixel sensor, the corresponding LIF neuron will issue a spike provided that the sum of excitatory post-synaptic potential (EPSP) caused by flow and receptive synapses exceeds the neuron's threshold. In this case the spike potential propagates, in a decaying α -function mode, to the flow synapse of its successor neuron. The estimated time when this successor neuron will fire, corresponding to the time taken for an edge to travel between sensors, is encoded in the time constant of an exponentially-decaying signal from the firing neuron. Effectively, the LIF neuron will fire only if the predicted time-of-arrival and the actual time-of-arrival are within a controlled time window. For simplicity we fix flow synapse efficacy while allowing the receptive synapse efficacy to adjust according to the online adaptation mechanism of spike-timing-dependent plasticity (STDP). The aim of this adaptation is to refine the "window" within which flow and receptive synapses must receive a signal to optimise both the recognition of real edges and the rejection of spurious edges. The numerical experiments are conducted in C++ by using Euler integration method and the update step is increased with each image frame.

Simulation results show that every neuron achieves an optimised window in signal propagation, i.e., only edges that arrive within a prescribed, specific timing window will activate the corresponding neuron. The results further demonstrate that, after an initial self-organising stage, the estimated and actual depth information agree well in both artificial and real images. Furthermore, receptive synapse weights converge from an initial random distribution towards a bimodal distribution. As our aim is to implement the LIF neuron in aVLSI, our model is therefore amenable to CMOS hardware and moves towards a demonstration of an autonomous on-chip adaptation procedure, based upon biological observations, with meaningful applications.

References

- [1] Dayan, P. and Abbott, L.F.: "Theoretical Neuroscience - Computational and Mathematical Modeling of Neural Systems", MIT Press, 2001.
- [2] Markram, H., Lubke, J., et al.: "Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs", *Science* **275**:213-215, 1997.
- [3] Wörgötter, F., Cozzi, A. and Gerdes V.: "A parallel noise-robust algorithm to recover depth information from radial flow fields", *Neural Computation* **11**:381-416, 1999.
- [4] Bofill, A. and Murray, A.F.: "Learning temporal correlations in biologically-inspired aVLSI", *IEEE International Symposium on Circuits and Systems*, Vol.5, pp.817-820, 2003.
- [5] Yang, Z. and Murray, A.F.: "A Biologically Plausible Neuromorphic System for Object Recognition and Depth Analysis", submitted to *European Symposium on Artificial Neural Networks (2004)*. Brugge, Belgium.

Appendix B

Component Sizes and Bias Voltages

Unless otherwise stated the size of a PMOS is $W = 2\mu\text{m}$, $L = 1\mu\text{m}$ and of an NMOS is $W = 1\mu\text{m}$, $L = 1\mu\text{m}$. All transistor dimensions are in microns.

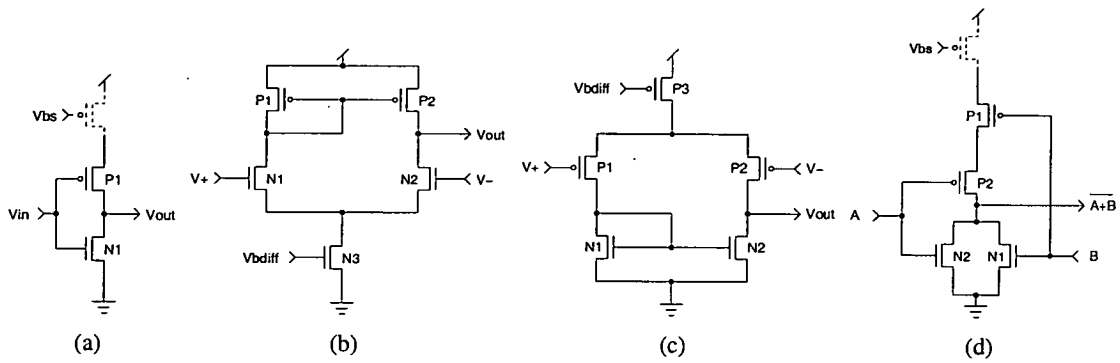


Figure B.1: Standard Circuits

If an inverter symbol is used in a circuit schematic it will look like B.1(a), without the transistor drawn with dashed lines, and the dimensions will be listed WP1, WN1 and LP1, LN1. If the inverter is marked *S* indicating that it has a slow rise time, the extra transistor is included and the dimensions WS and LS will be added. The bias voltage will also be specified. For the differential pair, figure B.1(b), the dimensions will be listed WP1, WP2, WN1, WN2, WN3 and LP1, LP2, LN1, LN2, LN3. V_b , unless otherwise stated is 700mV. For the alternative configuration, figure B.1(c), the dimensions will be listed WN1, WN2, WP1, WP2, WP3 and LN1, LN2, LP1, LP2, LP3. Unless stated the bias for this configuration is 2.5V. For the NOR gate the dimensions will be listed as WN1, WN2, WP1, WP2 and LN1, LN2, LP1, LP2. The dashed transistor is again only applicable if the NOR symbol is marked by an *S* and then the same conventions apply.

Figure 3.2

Component or Bias	Value
N1	W=3 L=3
N2	W=3 L=3
N4	W=3 L=3
Cm	3pF
Cfb	500fF
D1	type b, Vbdiff=700mV
Vleak	300mV
Vpw	1.2V
Vref	250mV

Figure 3.3

Component or Bias	Value
Vwfc	1.4V
Vleak	380mV

Figure 3.4

Component or Bias	Value
Vwf	2.5V

Figure 4.1

Component or Bias	Value
C1	1pF
C2	1pF
D1	type b, Vbdiff=700mV
Vb	2.85V

Figure 4.3

Component or Bias	Value - chip1 simulation	Value - chip2 fabrication
N2	W=0.6 L=0.3	W=0.4 L=0.35
N3	W=0.6 L=0.3	W=0.4 L=0.35
P2	W=0.6 L=0.3	W=0.4 L=0.35
P3	W=0.6 L=0.3	W=0.4 L=0.35
C1	1pF	1pF
C2	1pF	1pF
D1	type b, Vbdiff=700mV	type b, Vbdiff=700mV
Vb	2.87V	2.79V
Vrampth	0.5V	0.5V

Figure 4.5

Component or Bias	Value
N1	W=3 L=3
N3	W=3 L=3
P1	W=6 L=3
P2	W=6 L=3
P3	W=6 L=3
P4	W=6 L=3
C1	1pF
D1	type b, Vbdiff=700mV
Vb	280mV
Vdd_mirr	3.28 - 3.31V
Vrampth	0.5V

Figure 5.2

Component or Bias	Value
N1	W=3 L=30
N2	W=30 L=3
N3	W=3 L=3
N4	W=3 L=3
N5	W=30 L=3
N6	W=3 L=30
N7	W=30 L=3
P1	W=6 L=3
P2	W=6 L=3
P3	W=6 L=3
P4	W=6 L=3
Vcas	1.365V ¹
Vsynth	1.8V

¹Generated by an inverter with its output connected to the input. W = 4, 3 L = 3, 3.

Figure 5.4

Component or Bias	Value
N4	W=3 L=3
P1	W=6 L=3
C1	1pF
C2	1pF
C3	1pF
C4	1pF
Cw	1pF
Vb1	280mV
Vb2	2.90V
Vb3	3.2V
Vb4	280mV
Vn	800mV
Vp	2.3V

Figure 6.1

Component or Bias	Value
N1	W=0.6 L=0.35
N2	W=0.6 L=0.35
N3	W=0.6 L=0.35
N4	W=0.6 L=0.35
N7	W=0.4 L =0.35
P1	W=6 L=3
P2	W=6 L=3
P3	W=6 L=3
P4	W=6 L=3
P5	W=0.4 L=0.35
P6	W=0.4 L=0.35
I1	W=0.4, 0.4 L=0.35, 0.35
C1	1pF
Cslope	5.77pF
D1	type b, Vbdiff=700mV
Vb	325mV
Vdd_mirr	3.28 - 3.31V ²
Vrampth	0.5V
Select_PS	3.3 or 0V
Adapt_off	3.3 or 0V

²While testing the chip the range was dropped to 3.25 - 3.26V suggesting that there was a voltage drop along the Vdd power line.

Figure 6.2

Component or Bias	Value
N1	W=0.4 L=0.35
N2	W=0.4 L=0.35
N3	W=0.4 L=0.35
N4	W=0.4 L=0.35
N5	W=0.4 L=0.35
N6	W=0.4 L=0.35
N7	W=0.4 L=0.35
N8	W=0.4 L=0.35
C0	Moscap W=0.8 L=1
C1	4fF
C2	4fF
C3	4fF
Cslope	5.77pF
X1	WS=6 LS=6, VbS=2.7V
D1	type c, W=1, 1, 10, 10, 2 L=1, 1, 5, 5, 1, Vbdiff=2.5V
D2	type c, W=1, 1, 10, 10, 2 L=1, 1, 5, 5, 1, Vbdiff=2.5V
D3	type c, W=1, 1, 10, 10, 2 L=1, 1, 5, 5, 1, Vbdiff=2.5V
I2	WS=6 LS=6, VbS=2.7V
I4	WS=6 LS=6, VbS=2.7V
Vbp	2.5V
Adapt_off	3.3 or 0V

Figure 6.3

Component or Bias	Value
P1	W=1 L=5
P2	W=5 L=5
N1	W=7 L=5
N2	W=5 L=5
N3	W=7 L=5
N4	W=5 L=5
N5	W=20 L=5
N6	W=5 L=5
D1	type c, W=3, 3, 6, 6, 5, L=3, 3, 3, 3, 1, Vbdiff=2.5V
D2	type c, W=3, 3, 6, 6, 5, L=3, 3, 3, 3, 1, Vbdiff=2.5V
Vsfp	2.5V
Vsfn	200mV

Figure C.2

Component or Bias	Value
P1	W=0.8 L=0.35
P2	W=0.8 L=0.35
N1	W=3 L=3
N2	W=3 L=3
C1	1pF
C2	1pF
I1	W=0.8, 0.4, L=0.35, 0.35
I2	W=0.8, 0.4, L=0.35, 0.35
I3	WS=6 LS=3 VbS=2.5V
I4	W=10, 5, L=10, 10
I6	W=0.8, 0.4, L=0.35, 0.35
I7	W=0.8, 0.4, L=0.35, 0.35
I8	WS=6 LS=3 VbS=2.5V
I9	W=10, 5, L=10, 10
X1	W=0.8, 0.8, 0.4, 0.4, L=0.35, 0.35, 0.35, 0.35
X2	W=0.8, 0.8, 0.4, 0.4, L=0.35, 0.35, 0.35, 0.35
X3	W=0.8, 0.8, 0.4, 0.4, L=0.35, 0.35, 0.35, 0.35
X4	W=0.8, 0.8, 0.4, 0.4, L=0.35, 0.35, 0.35, 0.35
Vwinb	300mV

Appendix C

Control Circuits

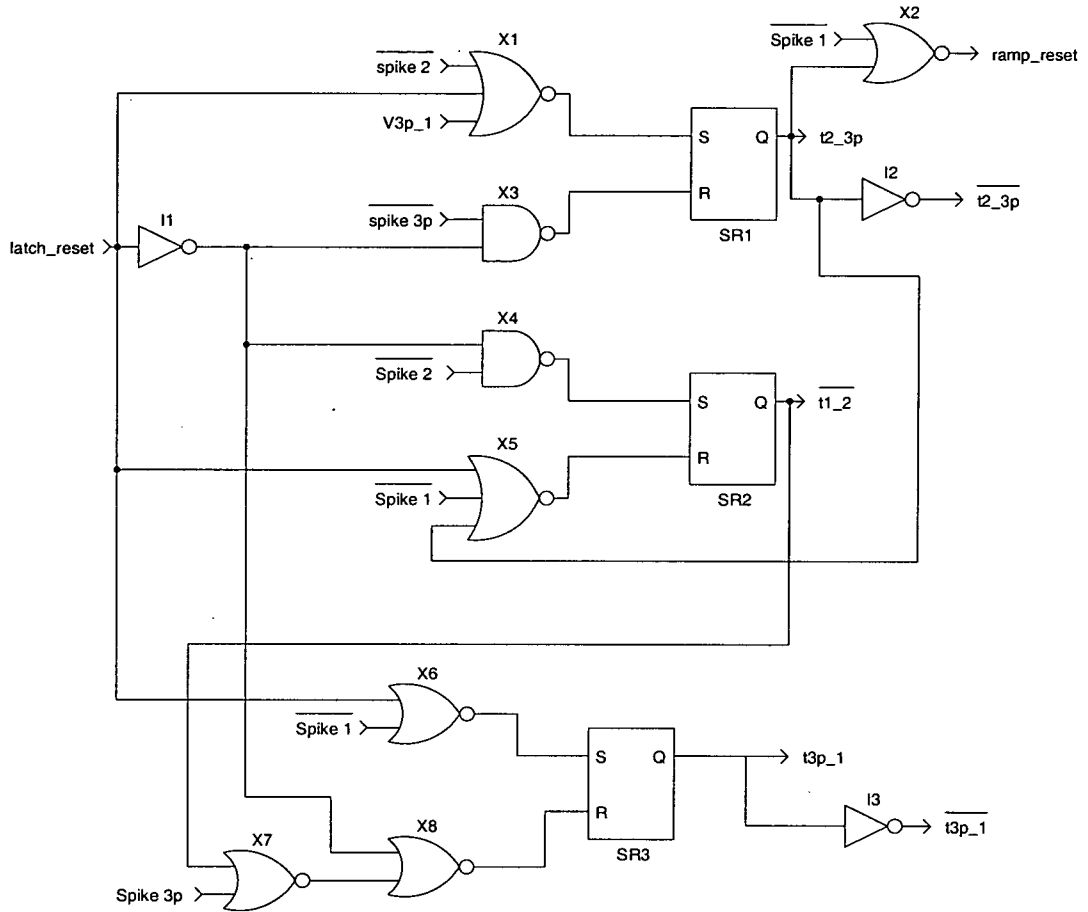


Figure C.1: Prediction Circuit Control Circuitry. The circuitry used to control the charging and discharging of the prediction capacitor for both prediction circuits described in chapter 6. Additional combinatorial logic is included before the latches to ensure control signals do not interfere with each other.

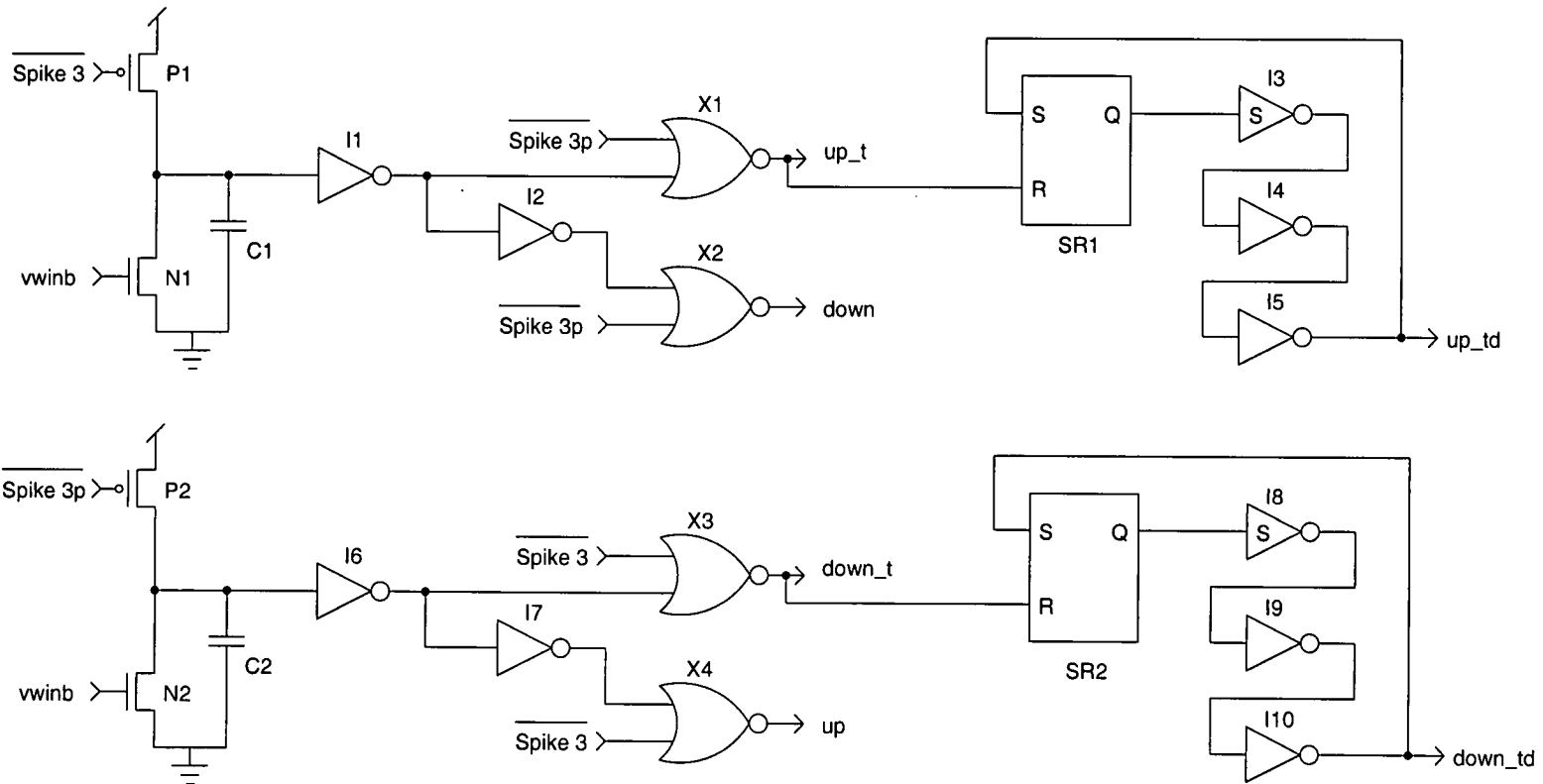


Figure C.2: Weight Change Control Circuitry. The circuitry used to generate the signals *up*, *up_t*, *up_td*, *down*, *down_t* and *down_td* for use in the circuit shown in figure 6.2.

References

- [1] F. Rieke, D. Warland, R. de Ruyter van Steveninck, and W. Bialek, *Spikes - Exploring the Neural Code*. MIT Press, 1997.
- [2] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, "Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs," *Science*, vol. 275, pp. 213–215, 1997.
- [3] G.-q. Bi and M.-m. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength and postsynaptic cell type," *Journal of Neuroscience*, vol. 18, no. 24, pp. 10 464–10 472, 15th December 1998.
- [4] L. I. Zhang, H. W. Tao, C. E. Holt, W. A. Harris, and M.-m. Poo, "A critical window for cooperation and competition among developing retinotectal synapses," *Nature*, vol. 395, pp. 37–44, 3rd September 1998.
- [5] R. E. Suri and T. J. Sejnowski, "Spike propagation synchronized by temporally asymmetric Hebbian learning," *Biological Cybernetics*, vol. 87, pp. 440–445, 2002.
- [6] W. Gerstner, R. Kempter, J. L. van Hemmen, and H. Wagner, "Hebbian learning of pulse timing in the barn owl auditory system," in *Pulsed Neural Networks*, W. Maass and C. M. Bishop, Eds. MIT Press, 1999, ch. 14, pp. 353–377.
- [7] C. W. Eurich, K. Pawelzik, U. Ernst, J. D. Cowan, and J. G. Milton, "Dynamics of self-organized delay adaptation," *Physical Review Letters*, vol. 82, no. 7, pp. 1594–1597, 15th February 1999.
- [8] W. Senn, M. Schneider, and B. Ruf, "Activity-dependent development of axonal and dendritic delays, or, why synaptic transmission should be unreliable," *Neural Computation*, vol. 14, no. 3, pp. 583–619, March 2002.
- [9] A. Bofill-i-Petit and A. F. Murray, "Synchrony detection and amplification by silicon neurons with STDP synapses," *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1296–1304, September 2004.
- [10] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 211–221, January 2006.
- [11] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching properties of MOS transistors," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1440, October 1989.
- [12] F. Forti and M. E. Wright, "Measurement of MOS current mismatch in the weak inversion region," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 2, pp. 138–142, February 1994.
- [13] P. R. Kinget, "Device mismatch and tradeoffs in the design of analog circuits," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 6, pp. 1212–1224, June 2005.

- [14] G. Wegmann and E. A. Vittoz, "Basic principles of accurate dynamic current mirrors," *IEEE Proceedings*, vol. 137, no. 2, pp. 95–100, April 1990.
- [15] J. Robert, P. Deval, and G. Wegmann, "Very accurate current divider," *Electronics Letters*, vol. 25, no. 14, pp. 912–913, 6th July 1989.
- [16] Y. S. Yee, L. M. Terman, and L. G. Heller, "A 1 mV MOS comparator," *IEEE Journal of Solid-State Circuits*, vol. 13, no. 3, pp. 294–297, June 1978.
- [17] S. A. Jackson, J. C. Killens, and B. J. Blalock, "A programmable current mirror for analog trimming using single-poly floating-gate devices in standard CMOS technology," *IEEE Transactions on Circuits and Systems—Part II: Analog and Digital Signal Processing*, vol. 48, no. 1, pp. 100–102, January 2001.
- [18] S. Shah and S. Collins, "A temperature independent trimmable current source," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. I, May 2002, pp. 713–716.
- [19] S.-C. Liu, J. Kramer, G. Indiveri, T. Delbrück, and R. Douglas, *Analog VLSI: Circuits and Principles*. MIT Press, 2002.
- [20] S. Shah, "Floating-gate based trimmable current sources," PhD Thesis, Oxford University, 2002.
- [21] F. Wörgötter, A. Cozzi, and V. Gerdes, "A parallel noise-robust algorithm to recover depth information from radial flow fields," *Neural Computation*, vol. 11, pp. 381–416, 1999.
- [22] E. D. Adrian, "The impulses produced by sensory nerve endings. Part 1," *Journal of Physiology*, vol. 61, no. 1, pp. 49–72, 1926.
- [23] E. D. Adrian and Y. Zotterman, "The impulses produced by sensory nerve endings. Part 2. The response of a single end-organ," *Journal of Physiology*, vol. 61, no. 2, pp. 151–171, 1926.
- [24] —, "The impulses produced by sensory nerve endings. Part 3. Impulses set up by touch and pressure," *Journal of Physiology*, vol. 61, no. 4, pp. 465–483, 1926.
- [25] W. Bialek, F. Rieke, R. R. de Ruyter van Steveninck, and D. Warland, "Reading a neural code," *Science*, vol. 252, pp. 1854–1857, 28th June 1991.
- [26] W. Gerstner, "Spiking neurons," in *Pulsed Neural Networks*, W. Maass and C. M. Bishop, Eds. MIT Press, 1999, ch. 1, pp. 3–53.
- [27] W. Gerstner and W. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.
- [28] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.
- [29] W. Maass and T. Natschläger, "Networks of spiking neurons can emulate arbitrary hopfield nets in temporal coding," *Network: Computation in Neural Systems*, vol. 8, no. 4, pp. 355–372, 1997.

-
- [30] B. Ruf and M. Schmitt, "Self-organizing maps of spiking neurons using temporal coding," in *Computational Neuroscience: Trends in Research, 1998*, J. M. Bower, Ed., Plenum Press, New York, 1998, pp. 509–514.
- [31] T. Natschläger and B. Ruf, "Online clustering with spiking neurons using temporal coding," in *Neuromorphic Systems: Engineering Silicon from Neurobiology*, L. S. Smith and A. Hamilton, Eds. World Scientific, 1998, pp. 33–42.
- [32] D. Z. Jin and H. S. Seung, "Fast computation with spikes in a recurrent neural network," *Physical Review E*, vol. 65, no. 051922, 2002.
- [33] W. Gerstner, R. Ritz, and J. L. van Hemmen, "Why spikes? Hebbian learning of time-resolved excitation patterns," *Biological Cybernetics*, vol. 69, pp. 503–515, 1993.
- [34] S. Thorpe, A. Delorme, and R. Van Rullen, "Spike-based strategies for rapid processing," *Neural Networks*, vol. 14, no. 6-7, pp. 715–725, 2001.
- [35] R. Van Rullen and S. J. Thorpe, "Rate coding versus temporal order coding: What the retinal ganglion cells tell the visual cortex," *Neural Computation*, vol. 13, pp. 1255–1283, 2001.
- [36] J. O'Keefe and M. L. Recce, "Phase relationship between hippocampal place units and the EEG theta rhythm," *Hippocampus*, vol. 3, no. 3, pp. 317–330, 1993.
- [37] W. E. Sullivan and M. Konishi, "Segregation of stimulus phase and intensity coding in the cochlear nucleus of the barn owl," *Journal of Neuroscience*, vol. 4, no. 7, pp. 1787–1799, 1984.
- [38] C. E. Carr and K. Konishi, "A circuit for detection of interaural time differences in the brain stem of the barn owl," *Journal of Neuroscience*, vol. 10, no. 10, pp. 3227–3246, 1990.
- [39] E. Vaadia, I. Haalman, M. Abeles, H. Bergman, Y. Prut, H. Slovin, and A. Aertsen, "Dynamics of neuronal interactions in monkey cortex in relation to behavioural events," *Nature*, vol. 373, pp. 515–518, 1995.
- [40] F. Rieke, D. A. Bodnar, and W. Bialek, "Naturalistic stimuli increase the rate and efficiency of information transmission by primary auditory afferents," *Proceedings of the Royal Society of London: B*, vol. 262, no. 1365, pp. 259–265, 1995.
- [41] D. O. Hebb, *The organization of behaviour*. Wiley, 1949.
- [42] T. H. Brown and S. Chattarji, "Hebbian synaptic plasticity," in *The Handbook of Brain Theory and Neural Networks*, M. A. Arbib, Ed. MIT Press, 1998, pp. 454–459.
- [43] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Prentice Hall, 1999.
- [44] S. Song, K. D. Miller, and L. F. Abbot, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neuroscience*, vol. 3, no. 9, pp. 919–926, September 2000.

-
- [45] C. C. Bell, V. Z. Han, Y. Sugawara, and K. Grant, "Synaptic plasticity in a cerebellum-like structure depends on temporal order," *Nature*, vol. 387, pp. 278–281, 15th May 1997.
- [46] V. Egger, D. Feldmeyer, and B. Sakmann, "Coincidence detection and changes of synaptic efficacy in spiny stellate neurons in rat barrel cortex," *Nature Neuroscience*, vol. 2, no. 12, pp. 1098–1105, December 1999.
- [47] D. Debanne, B. H. Gähwiler, and S. M. Thompson, "Long-term synaptic plasticity between pairs of individual CA3 pyramidal cells in rat hippocampal slice cultures," *Journal of Physiology*, vol. 507, no. 1, pp. 237–247, 1998.
- [48] D. E. Feldman, "Timing-based LTP and LTD at vertical inputs to layer II/III pyramidal cells in rat barrel cortex," *Neuron*, vol. 27, pp. 45–56, July 2000.
- [49] Y. Dan and M.-m. Poo, "Spike timing-dependent plasticity of neural circuits," *Neuron*, vol. 44, pp. 23–30, 30th September 2004.
- [50] P. D. Roberts and C. C. Bell, "Spike timing dependent synaptic plasticity in biological systems," *Biological Cybernetics*, vol. 87, pp. 392–403, 2002.
- [51] M. C. W. van Rossum, G. Q. Bi, and G. G. Turrigiano, "Stable Hebbian learning from spike timing-dependent plasticity," *Journal of Neuroscience*, vol. 20, pp. 8812–8821, 1st December 2000.
- [52] L. F. Abbott and K. I. Blum, "Functional significance of long-term potentiation for sequence learning and prediction," *Cerebral Cortex*, vol. 6, pp. 406–416, May/June 1996.
- [53] R. P. N. Rao and T. J. Sejnowski, "Predictive sequence learning in recurrent neocortical circuits," in *Advances in Neural Information Processing Systems*, vol. 12. MIT Press, 2000, pp. 164–170.
- [54] C. A. Boettiger and A. J. Doupe, "Developmentally restricted synaptic plasticity in a songbird nucleus required for song learning," *Neuron*, vol. 31, pp. 809–818, 13th September 2001.
- [55] S. Song and L. F. Abbott, "Cortical development and remapping through spike timing-dependent plasticity," *Neuron*, vol. 32, pp. 339–350, 25th October 2001.
- [56] T. Nowotny, V. P. Zhigulin, A. I. Selverston, H. D. I. Abarbanel, and M. I. Rabinovich, "Enhancement of synchronization in a hybrid neural circuit by spike-timing dependent plasticity," *Journal of Neuroscience*, vol. 23, no. 30, pp. 9776–9785, 29th October 2003.
- [57] W. Gerstner, R. Kempter, J. L. van Hemmen, and H. Wagner, "A neuronal learning rule for sub-millisecond temporal coding," *Nature*, vol. 383, pp. 76–78, 5th September 1996.
- [58] R. Kempter, C. Leibold, H. Wagner, and J. L. van Hemmen, "Formation of temporal-feature maps by axonal propagation of synaptic learning," *Proceedings of the National Academy of Sciences*, vol. 98, no. 7, pp. 4166–4171, 27th March 2001.
- [59] C. Mead, *Analog VLSI and Neural Systems*. Addison Wesley, 1989.

-
- [60] —, “Neuromorphic electronic systems,” *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–1636, October 1990.
- [61] E. Chicca, D. Badoni, V. Dante, M. D’Andreagiovanni, G. Salina, L. Carota, S. Fusi, and P. Del Giudice, “A VLSI recurrent network of integrate-and-fire neurons connected by plastic synapses with long-term memory,” *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 1297–1307, September 2003.
- [62] G. Indiveri, “Modeling selective attention using a neuromorphic analog VLSI device,” *Neural Computation*, vol. 12, no. 12, pp. 2857–2880, December 2000.
- [63] A. van Schaik, “Building blocks for electronic spiking neural networks,” *Neural Networks*, vol. 14, no. 6-7, pp. 617–628, 2001.
- [64] G. Indiveri, “A low-power adaptive integrate-and-fire neuron circuit,” in *Proc. IEEE International Symposium on Circuits and Systems*, vol. IV, May 2003, pp. 820–823.
- [65] R. Jolivet, A. Rauch, H.-R. Lüscher, and W. Gerstner, “Integrate-and-fire models with adaptation are good enough: Predicting spike times under random current injection,” in *Advances in Neural Information Processing Systems*, vol. 18. MIT Press, 2006, pp. 595–602.
- [66] A. L. Hodgkin and A. F. Huxley, “A quantitative description of membrane current and its application to conduction and excitation in nerve,” *Journal of Physiology*, vol. 117, pp. 500–544, 1952.
- [67] M. Mahowald and R. Douglas, “A silicon neuron,” *Nature*, vol. 354, pp. 515–518, 1991.
- [68] S. Saïghi, Y. Bornat, J. Tomas, and S. Renaud, “Neuromimetic ICs and system for parameters extraction in biological neuron models,” in *Proc. IEEE International Symposium on Circuits and Systems*, May 2006, pp. 4207–4211.
- [69] K. A. Boahen, “Retinomorph vision systems: reverse engineering the vertebrate retina,” PhD Thesis, California Institute of Technology, 1997.
- [70] C. Diorio, P. Hasler, B. A. Minch, and C. A. Mead, “A single-transistor silicon synapse,” *IEEE Transactions on Electron Devices*, vol. 43, no. 11, pp. 1972–1980, November 1996.
- [71] C. Diorio, D. Hsu, and M. Figueroa, “Adaptive CMOS: From biological inspiration to systems-on-a-chip,” *Proceedings of the IEEE*, vol. 90, no. 3, pp. 345–357, March 2002.
- [72] S.-C. Liu, M. Boegershausen, and P. Suter, “Circuit model of short-term synaptic dynamics,” in *Advances in Neural Information Processing Systems*, vol. 15. MIT Press, 2003, pp. 1075–1082.
- [73] L. F. Abbott, J. A. Varela, K. Sen, and S. B. Nelson, “Synaptic depression and cortical gain control,” *Science*, vol. 275, pp. 220–224, 1997.
- [74] M. V. Tsodyks and H. Markram, “The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability,” *Proceedings of the National Academy of Sciences*, vol. 94, pp. 719–723, 1997.

- [75] G. Indiveri, "Neuromorphic bistable VLSI synapses with spike-timing-dependent plasticity," in *Advances in Neural Information Processing Systems*, vol. 15, 2003, pp. 1115–1122.
- [76] P. Häfliger and H. K. Riis, "A multi-level static memory cell," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. I, May 2003, pp. 25–28.
- [77] C.-K. Sin, A. Kramer, V. Hu, R. R. Chu, and P. K. Ko, "EEPROM as an analog storage device, with particular applications in neural networks," *IEEE Transactions on Electron Devices*, vol. 39, no. 6, pp. 1410–1419, June 1992.
- [78] C. Koch, *Biophysics of Computation*. Oxford University Press, 1999.
- [79] R. J. Vogelstein, F. Tenore, R. Philipp, M. S. Adlerstein, D. H. Goldberg, and G. Cauwenberghs, "Spike timing-dependent plasticity in the address domain," in *Advances in Neural Information Processing Systems*, vol. 15. MIT Press, 2003, pp. 1147–1154.
- [80] D. H. Goldberg, G. Cauwenberghs, and A. G. Andreou, "Analog VLSI spiking neural network with address domain probabilistic synapses," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. III, May 2001, pp. 241–244.
- [81] J. V. Arthur and K. Boahen, "Learning in silicon: Timing is everything," in *Advances in Neural Information Processing Systems*, vol. 18. MIT Press, 2006, pp. 75–82.
- [82] K. A. Boahen, "Communicating neuronal ensembles between neuromorphic chips," in *Neuromorphic Systems Engineering: Neural Networks in Silicon*, T. S. Lande, Ed. Kluwer Academic Publishers, Norwell, MA, 1998, ch. 11, pp. 229–259.
- [83] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gómez-Rodríguez, H. K. Riis, T. Delbrück, S. C. Liu, S. Zahnd, A. M. Whatley, R. Douglas, P. Häfliger, G. Jimenez-Moreno, A. Civit, T. Serrano-Gotarredona, A. Acosta-Jiménez, and B. Linares-Barranco, "AER building blocks for multi-layer multi-chip neuromorphic vision systems," in *Advances in Neural Information Processing Systems*, vol. 18. MIT Press, 2006, pp. 1217–1224.
- [84] R. F. Lyon and C. Mead, "An analog electronic cochlea," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1119–1134, July 1988.
- [85] A. van Schaik and S.-C. Liu, "AER EAR: A matched silicon cochlea pair with address event representation interface," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. V, May 2005, pp. 4213–4216.
- [86] R. Sarpeshkar, C. Salthouse, J.-J. Sit, M. W. Baker, S. M. Zhak, T. K. T. Lu, L. Turicchia, and S. Balster, "An ultra-low-power programmable analog bionic ear processor," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 4, pp. 711–727, April 2005.
- [87] J. H. Wittig Jr and K. Boahen, "Silicon neurons that phase-lock," in *Proc. IEEE International Symposium on Circuits and Systems*, May 2006, pp. 4535–4538.
- [88] K. A. Zaghloul and K. Boahen, "Optic nerve signals in a neuromorphic chip," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 4, pp. 657–675, April 2004.

- [89] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128x128 120dB 30mw asynchronous vision sensor that responds to relative intensity change," in *International Solid State Circuits Conference Digest of Technical Papers*, February 2006, pp. 414–415.
- [90] J. Kramer, "An integrated optical transient sensor," *IEEE Transactions on Circuits and Systems—Part II: Analog and Digital Signal Processing*, vol. 49, no. 9, pp. 612–628, September 2002.
- [91] S.-C. Liu, J. Kramer, G. Indiveri, T. Delbrück, T. Burg, and R. Douglas, "Orientation-selective aVLSI spiking neurons," *Neural Networks*, vol. 14, no. 6-7, pp. 629–643, 2001.
- [92] G. Indiveri, R. Mürer, and J. Kramer, "Active vision using an analog VLSI model of selective attention," *IEEE Transactions on Circuits and Systems—Part II: Analog and Digital Signal Processing*, vol. 48, no. 5, pp. 492–500, May 2001.
- [93] X. Qi, X. Guo, and J. G. Harris, "A time-to-first spike CMOS imager," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. IV, May 2004, pp. 824–827.
- [94] P. Häfliger and E. J. Aasebø, "A rank encoder: Adaptive analog to digital conversion exploiting time domain spike signal processing," *Analog Integrated Circuits and Signal Processing*, vol. 40, pp. 39–51, 2004.
- [95] P. Häfliger, M. Mahowald, and L. Watts, "A spike based learning neuron in analog VLSI," in *Advances in Neural Information Processing Systems*, vol. 9. MIT Press, 1997, pp. 692–698.
- [96] T. Kohonen, *Self-Organization and Associative Memory*. Springer, 1984.
- [97] C. Gordon and P. Hasler, "Biological learning modeled in an adaptive floating-gate system," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. V, May 2002, pp. 609–612.
- [98] A. Bofill, A. F. Murray, and D. P. Thomson, "Circuits for VLSI implementation of temporally-asymmetric Hebbian learning," in *Advances in Neural Information Processing Systems*, vol. 14. MIT Press, 2002, pp. 1091–1098.
- [99] A. Bofill-i-Petit and A. F. Murray, "Synchrony detection by analogue VLSI neurons with bimodal STDP synapses," in *Advances in Neural Information Processing Systems*, vol. 16. MIT Press, 2004.
- [100] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI reconfigurable network of integrate-and-fire neurons with spike-based learning synapses," in *Proc. European Symposium on Artificial Neural Networks*, April 2004, pp. 405–410.
- [101] K. R. Lakshmikumar, R. A. Hadaway, and M. A. Copeland, "Characterization and modeling of mismatch in MOS transistors for precision analog design," *IEEE Journal of Solid-State Circuits*, vol. 21, no. 6, pp. 1057–1066, December 1986.
- [102] M. J. M. Pelgrom, H. P. Tuinhout, and M. Vertregt, "Transistor matching in analog CMOS applications," in *Proc. International Electron Devices Meeting (IEDM'98)*, December 1998, pp. 915–918.

- [103] K. Bult, "Analog design in deep sub-micron CMOS," in *Proceedings of the 26th European Solid-State Circuits Conference*, September 2000, pp. 126–132.
- [104] A. Hastings, *The Art of Analog Layout*. Prentice Hall, 2001.
- [105] S. J. Lovett, M. Welten, A. Mathewson, and B. Mason, "Optimizing MOS transistor mismatch," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 1, pp. 147–150, January 1998.
- [106] P. G. Drennan and C. C. McAndrew, "Understanding MOSFET mismatch for analog design," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 3, pp. 450–456, March 2003.
- [107] A.-J. Annema, B. Nauta, R. van Langevelde, and H. Tuinhout, "Analog circuits in ultra-deep-submicron CMOS," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 132–143, January 2005.
- [108] A. Pavasović, A. G. Andreou, and C. R. Westgate, "Characterization of subthreshold MOS mismatch in transistors for VLSI systems," *Analog Integrated Circuits and Signal Processing*, vol. 6, pp. 75–85, July 1994.
- [109] E. Vittoz and J. Fellrath, "CMOS analog integrated circuits based on weak inversion operations," *IEEE Journal of Solid-State Circuits*, vol. 12, no. 3, pp. 224–231, June 1977.
- [110] E. A. Vittoz, "The design of high-performance analog circuits on digital CMOS chips," *IEEE Journal of Solid-State Circuits*, vol. 20, no. 3, pp. 657–665, June 1985.
- [111] J. Bastos, A. M. Marques, M. S. J. Steyaert, and W. Sansen, "A 12-bit intrinsic accuracy high-speed CMOS DAC," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 12, pp. 1959–1969, December 1998.
- [112] J. Maunu, M. Pänkäälä, J. Marku, J. Poikonen, M. Laiho, and A. Paasio, "Current source calibration by combination selection of minimum sized devices," in *Proc. IEEE International Symposium on Circuits and Systems*, May 2006, pp. 549–552.
- [113] D. W. J. Groeneveld, H. J. Schouwenars, H. A. H. Termeer, and C. A. A. Bastiaansen, "A self-calibration technique for monolithic high-resolution D/A converters," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 6, pp. 1517–1522, December 1989.
- [114] S. J. Daubert, D. Vallancourt, and Y. P. Tsividis, "Current copier cells," *Electronics Letters*, vol. 24, no. 25, pp. 1560–1562, 8 December 1988.
- [115] C. Taillefer and C. Wang, "Current mirror compensation for transistor mismatch," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. II, May 2000, pp. 509–512.
- [116] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*, International ed. McGraw-Hill, 1995.
- [117] M. A. Dahlem and F. Wörgötter, "Rotation-invariant optical flow by gaze-dependent retino-cortical mapping," in *Proceedings of the 2nd International Workshop on Biologically Motivated Computer Vision*, ser. Lecture Notes in Computer Science, vol. 2525, 2002, pp. 137–145.

-
- [118] A. Bofill-i-Petit and A. F. Murray, "Learning temporal correlations in biologically-inspired aVLSI," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 5, May 2003, pp. 817–820.
 - [119] S. Loeda, "Circuits to support antisymmetric Hebbian training in artificial neural networks," BEng Hons Thesis, The University of Edinburgh, May 2002, available at <http://www.sebastian.loeda.com>.
 - [120] K. Cameron, V. Boonsobhak, A. Murray, and D. Renshaw, "Spike timing dependent plasticity (STDP) can ameliorate process variations in neuromorphic VLSI," *IEEE Transactions on Neural Networks*, vol. 16, no. 6, pp. 1626–1637, November 2005.
 - [121] R. J. Vogelstein, U. Mallik, and G. Cauwenberghs, "Silicon spike-based synaptic array and address-event transceiver," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. V, May 2004, pp. 385–388.
 - [122] E. A. Vittoz and G. Wegmann, "Dynamic current mirrors," in *Analogue IC design: the current-mode approach*, C. Toumazou, F. J. Lidgey, and D. G. Haigh, Eds. Peter Peregrinus Ltd. on behalf of the Institution of Electrical Engineers, Reprint Edition, 1998, ch. 7, pp. 297–326.
 - [123] G. Cauwenberghs, "Analog VLSI stochastic perturbative learning architectures," *Analog Integrated Circuits and Signal Processing*, vol. 13, no. 1/2, pp. 195–209, 1997.